Aalto University
School of Electrical
Engineering

LULEÅ
UNIVERSITY
OF TECHNOLOGY

Nemanja Jovanović

Aalto-2 satellite attitude control system

School of Electrical Engineering

Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Technology.

Espoo, 18.8.2014

Thesis supervisors:

Professor Emeritus Aarne Halme

Professor Thomas Gustafsson

Thesis instructor:

Msc. Osama Khurshid

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Nemanja Jovanović

Title: Aalto-2 satellite attitude control system

Date: 18.8.2014 Language: English Number of pages: 6+64

Department of Electrical Engineering and Automation

Professorship: Automation Techonology Code: AUT-84

Supervisors: Professor Emeritus Aarne Halme, Professor Thomas Gustafsson

Instructor: Msc. Osama Khurshid

The Attitude Control System for the Aalto-2 satellite was designed and verified if it meets the requirements of the QB50 project mission. Attitude control is achieved through the passive atmospheric drag torque stabilization and active control over magnetorquers. For the detumble phase, $\dot{B}$ control is used, and for nominal stabilization phase, PD, LQR and SDRE control methods were investigated and compared, through the computer simulations. Software algorithms for solving of Algebraic Matrix Riccati Equation (Schur decomposition and Kleinman) and system tasks were realized for the onboard computer and the overall system functionality was tested with the Hardware In the Loop method. LQR control method showed the best performance, though none of the controllers completely met the mission needs.

Keywords: Attitude Control System, B dot, Linear Quadratic Regulator, State Dependent Riccati Equation, PD controller, Simulation, Hardware In the Loop

Preface

This thesis is the concluding project of the international master program in Space Science and Technology, SpaceMaster. The project was done at the department of Electrical Engineering and Automation of Aalto University, as a part of the Aalto-2 satellite project. The Aalto-2 satellite is the second CubeSat form satellite developed at the university by students.

First of all, I would like to thank the Luleå University for awarding me with the scholarship, which enabled me to attend this master program. I would also like to thank my instructor Osama Khurshid for his excelent guidance, Prof. em. Aarne Halme for his comprehesive feedback and Lic.Sc.(Tech.) Tomi Ylikorpi for all the help he provided during the studies at the Aalto University. Finally, special thanks to all the Round 8 SpaceMaster students for the friendship and all the great times we had during the past two years.

Otaniemi, 12.8.2014

Nemanja Jovanović

Contents

Symbols and abbreviations

Symbols

A	System Matrix
B	Input Matrix
$\mathbf{B}$	Magnetic field
c	Speed of light
E_k, E_p	Kinetic and Potential energies
F_s	Solar energy flux
$\mathcal{F}_i$, $\mathcal{F}_o$, $\mathcal{F}_b$	Inertial, Orbit and Body reference frames
I, I_x, I_y, I_z	Moment of Inertia matrix and its diagonal components
λ	Matrix eigenvalues
$\mathbf{m}$	Magnetic dipole
μ	Earth's gravitational constant
$\mathbf{q}$, $\overline{\mathbf{q}}$, $\mathbf{q}_e$, $\mathbf{q}_d$, $\mathbf{q}_c$, $\mathbf{q}_{ob}$	Unit quaternion, its vector part, attitude error, desired attitude, current attitude, body to orbit reference frame rotation
R_e	Distance between the center of the Earth and Satellite
R_o^b	DCM rotation matrix from orbit to body reference frame
T, T_c, T_g, T_d	Total, control, gravity gradient and atmospheric drift torques
u	Input to the system
$\mathbf{V}_r$	Satellite's velocity vector
$\omega_{\mathbf{ob}}^{\mathbf{b}}$, $\omega_{\mathbf{ib}}^{\mathbf{b}}$, $\omega_{\mathbf{io}}^{\mathbf{o}}$	Angular velocity vectors, body relative to orbit in body frame, body relative to inertial in body frame and orbit relative to inertial in orbit frame
ω_0	Orbit angular velocity
x	System state

Abbreviations

ACS	Attitude Control System
ADS	Attitude Determination System
AMRE	Algebraic Matrix Riccati Equation
DCM	Direction Cosine Matrix
FLOP	Floating-point Operations
HIL	Hardware In the Loop
IGRF	International Geomagnetic Reference Field
LQR	Linear Quadratic Regulator
PD	Proportional-Derivative
RTOS	Real Time Operating System
SDC	State Dependent Coefficients
SDRE	State Dependent Riccati Equation
SGP	Simplified General Perturbations
TLE	Two-Line Elements

1 Introduction

The introduction of the CubeSat standard have created plenty of opportunities for affordable orbit deployment of pico-satellites. It is easy to attach them to most of the launch vehicles due to their small size and weight, thus sharing the costs with the main payload. All CubeSats tend to have similar structures, because of the defined dimensions and other CubeSat standard requirements, which further reduces the development costs. Designs are shared for commonly used parts and availability of often used components increase, raising the Technology Readiness Level. Also, many common problems for pico-satellites get solved and new uses are being discovered. All of this produces a 'technological ecosystem' that drives the research in the field of small-factor satellites and their use, often carried by the students and young experts.

1.1 The QB50 Project

The QB50 project is envisioned as an attempt to use many CubeSat form satellites in order to conduct an affordable in-situ research of the largely unexplored Earth's lower thermospehere. This region of the atmosphere, which lies between 200 and 380 kilometers altitude, is dense enough that the atmospheric drag effect is significant. This effectively limits the duration of the missions to several months. Large and expensive satellites can hardly justify their cost for orbits within this region. Sounding rockets are also able to position the instruments in the lower thermosphere, however only for short periods of time. And the ranged sensing with the radars can produce only scarce data due to the high transparency of the lower thermospehere to all of the frequency bands. However, deployment of CubeSat satellites in the string-of-pearls formation can provide longer time of multipoint in-situ measurements at affordable costs.

There are four main goals of QB50 project:

Facilitating Access to Space Achieving sustained and affordable access to space for small scale research missions by increasing the Technology Readiness Level for space applications. Defining new and improving existing standards for launch vehicles and CubeSat satellites, to better suit missions of this kind.

Scientific Research Obtaining of the scientific in-situ measurements of lower thermosphere. Collected data should improve the knowledge and models of the Earth's atmosphere. Most of the satellites will carry one of three sensor sets. Each of the sets contains either Ion-Neutral Mass Spectrometer, Flux-Φ-Probe or multi-Needle Langmuir Probe. Additionally, all of the sensor sets are provided with temperature measuring sensors.

In-Orbit Demonstration Some of the satellites, instead of carrying one of the sensor sets, will be used for testing and demonstration of novel technological systems for small satellites.

Education Designing and building of the satellites is delegated to universities. Thus allowing young engineers and students to learn about space engineering and gain valuable practical research and work experience.

1.2 Attitude Control System

Attitude Control System is a system that is in charge of maintaining the satellite's orientation. It can do so by passive or active methods. Passive methods elevate the environment's conditions by including specific design choices. For instance, arranging the mass distribution and adding of the long boom can make the satellite's orientation susceptible to the gravitational force, static magnet will try to align with the Earth's magnetic field, and aerodynamic design can affect orientation at lower altitudes where the atmospheric density is high. Active methods use actuators to adjust the satellite's attitude in an automated manner, for which they usually need the knowledge of the current attitude. Passive and active methods can sometimes be combined.

There exist several different actuators for control of satellite's attitude. However, the attitude control through the magnetic torques receives increasing attention for small form satellites. Other actuators for generating torques are not as suitable because of limited size of satellites. Thrusters require a fuel reservoirs which take up a lot of space and have a limited operational time. Reaction and momentum wheels can come in different sizes, respectful to the range of torques they can generate and their saturation limits. Still, there is a need for unloading of their moments by magnetic torquers, and having both makes the system more complex. On the other hand, a satellite with only magnetic actuators is simple in design, it takes up less space and is light. Magnetic actuators do not have any moving parts, which makes them quite robust. Their cost is much lower than the cost of other more sophisticated actuators. Unfortunately, the magnetic actuators are dependent on the earth's magnetic field. They can generate torque only in the plane normal to the magnetic field lines. This limitation introduces some difficulties for the attitude control.

Various three-axis attitude control approaches are used on the pico-satellites with magnetic actuators, and more are being researched. Depending on the satellites mission and requirements some control strategies perform better than the others. Several control methods will be described and analyzed. Purpose is to select and design appropriate Attitude Control System for Aalto-2 satellite. This will be done with computer and Hardware In the Loop (HIL) simulations for validation and verification of the system algorithms and hardware implementation.

2 Mission Details and Requirements

Aalto-2 satellite will be one of fifty satellites deployed in QB50 project. It will be a double-unit CubeSat equipped with the sensor set containing the multi-Needle Langmuir Probe, and its role will be collection of the measurements data in its own orbit. Langmuir probe sensor requires that the satellite keeps its attitude in a way that the probes are in front of the satellite, in the direction of the satellites movement. This requirement ensures that the probes are not in the wake of satellite, and thus not sheltered from the atmosphere environment.

Satellite will be deployed in a Sun-synchronous circular orbit. Altitude is expected to be between 350 and 400 kilometers. Orbit's inclination will be $98.6 \pm 0.08°$ and eccentricity ± 0.04. Due to the atmospheric drag effect, altitude of the orbit will decrease in time. Mission time is expected to last for around three months, when the satellite will deorbit after a significant loss of altitude.

The mission consists of several phases:

1. Satellite will be activated upon its deployment in orbit and the first phase is bringing up all of the systems and making sure they are working properly. Probes and antennas will be also deployed.

2. After the satellite's deployment in orbit, it might have large initial angular velocity. In second phase Attitude Control System shall slow it down. Detumble process is required to be able to decelerate tip-off rates of $10°/sec$ within two days to $0.8°/sec$. During this phase only electrical power, onboard computer, communication, attitude determination and control systems will be active. (Singarayar et al., 2013)

3. Third phase is a nominal phase which lasts until the end of the mission. In this phase the satellite's attitude should be maintained in a correct orientation. Multi-Needle Langmuir Probe will be attached on the ram side of the satellite, along the minor inertial axis of the body. Because of this, satellite will have a proper attitude for conducting the measurements when its minor inertial axis is parallel to the satellite's velocity vector. Sensor set requires the pointing error to be less than 15°. All of the system are active in this phase

4. Safe mode is a fourth phase, which is foreseen in case of problems and malfunctions. One possible reason to switch to the safe mode is if the battery falls below 60% capacity. This mode turns off the attitude control system and restricts telemetry transfers only to beacon signals.

3 Definitions and Satellite Description

3.1 Frames of reference

Satellite's orientation and position in orbit are described using frames of reference. All reference frames are right orthogonal coordinate systems. Three frames are of interest for attitude control: inertial, orbit and body reference frames. They are shown on Figure 1.

Inertial frame of reference ($\mathcal{F}_i$) Inertial frames are frames that do not sense any acceleration. The center of this frame is in the center of the Earth, with its z axis pointing to the north pole. x axis is parallel to the line of vernal equinox and points in direction from Sun to the Earth. And y axis completes the right hand coordinate system.

Orbit frame of reference ($\mathcal{F}_o$) The orbit reference frame have its center at the satellites center of mass. Toward the center of Earth points the z axis. Normal to it in orbits plane, and in direction of the satellites movement, is x axis. Again, y axis completes the right hand coordinate system.

Body frame of reference ($\mathcal{F}_b$) The center of this frame of reference is in the satellites center of mass and its axes coincide with the body's principal axes of inertia. If the body have uneven principal moments of inertia, it is considered that the major moment of inertia axis is parallel to the body's z axis and minor moment of inertia axis is parallel to the body's x axis. The frame is fixed to the body and co-rotates with it.

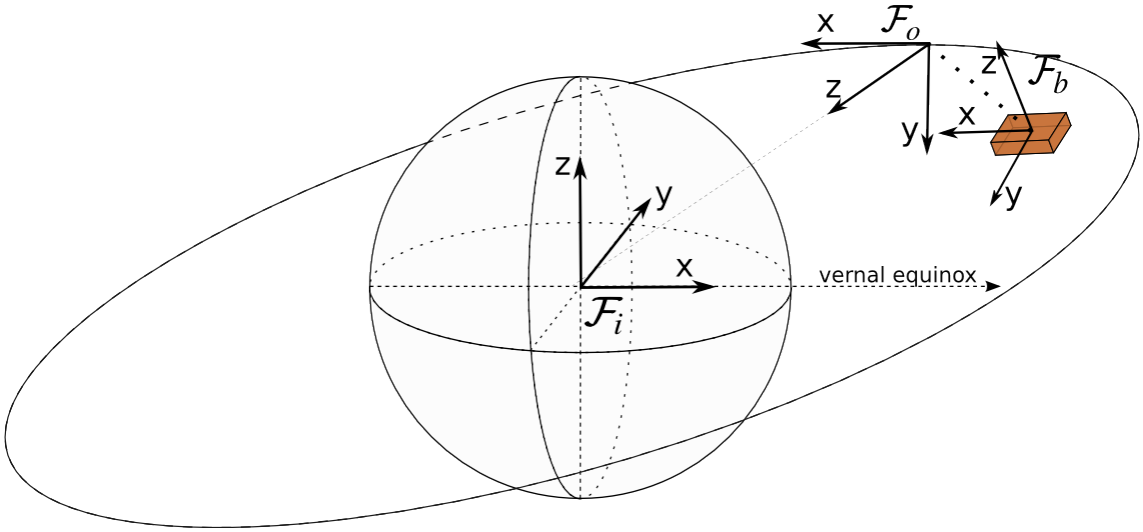

Figure 1: Frames of reference

3.2 Unit Quaternions

There are multiple ways to represent a rotation between the frames. Three dimensional rotation group, $SO(3)$, is often used as a direct representation, as it defines all possible rotations about the origin. However, this representation requires a Direction Cosine Matrix (DCM) with nine elements, which are redundant. A number of different representations with less elements have been defined, like Euler parameters and Euler angles. Another representation, unit quaternions (also called Euler symmetric parameters) are quite popular, for they avoid singularities, require only four parameters and conversion between them and Direction Cosine Matrix can be done without trigonometric operations. However, unit quaternions cover the rotation space $SO(3)$ twice, providing two quaternion values for any possible attitude of the satellite's body. This property of the unit quaternions can result in unwanted discontinuities in the systems dynamics and unwinding phenomenon (Bhat and Bernstein, 2000) if there is no mechanism for selecting the right unit quaternion value.

The unit quaternion is defined in four dimensional vector space, $\mathbf{q} \in \Re^4$, and will be denoted as:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{1}$$

The values q_1, q_2 and q_3 are called the vector part of the unit quaternion. The vector part of the unit quaternion will also be written as $\overline{\mathbf{q}} = [q_1 \; q_2 \; q_3]^T$. The last value, q_4, is a scalar part. Connection of the unit quaternion parameters with the rotation vector and amount of rotation around it is given as

$$\begin{aligned} q_1 &= e_1 sin\frac{\alpha}{2} \\ q_2 &= e_2 sin\frac{\alpha}{2} \\ q_3 &= e_3 sin\frac{\alpha}{2} \\ q_4 &= cos\frac{\alpha}{2} \end{aligned} \tag{2}$$

The values e_1, e_2 and e_3 define the unit vector of rotation, and α is the angle of rotation. Also the relation $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$ always holds.

Important operation with the unit quaternions is the relative orientation between two of them. With this operation the error between the desired and current attitude can be calculated. Result is also in a form of the unit quaternion.

$$\mathbf{q}_e = \mathbf{q}_d^{-1} \odot \mathbf{q}_c \tag{3}$$

$\mathbf{q}_d$ is a desired attitude and $\mathbf{q}_c$ is a current attitude, $\mathbf{q}_e$ represents the relative attitude between those two quaternions and $\odot$ is a quaternion multiplication.

Double covering of the rotation space by the quaternions can be expressed with the equation

$$\mathbf{q} = -\mathbf{q} \tag{4}$$

This may pose a problem when the unit quaternion is used as the error in inconsistent state feedback control system (Mayhew et al., 2011). Whichever unit quaternion of the two, 'positive' or 'negative', is calculated, the value is correct, and the system state will be guided toward the goal orientation. However, the path toward the goal state might be shorter or longer. Obviously, the shorter path to goal is preferred.

Unwinding phenomenon happens when the state is near the goal attitude, and the unit quaternion is changed to guide the system through the longer path, forcing the system to completely turn around and reach the same attitude.

To prevent the unwinding phenomenon and to ensure the selection of the shortest path toward the goal attitude there is a need for a mechanism to select the appropriate unit quaternion value. Mayhew et al. (2011) presents two such mechanisms. Memoryless and hybrid dynamic. The memoryless approach is chosen for its simplicity, though there is possibility for stabilizing in an attitude far from the goal. This possibility arises only in a small subset of the state space, which makes it improbable to happen. And even if this stability is reached, disturbances and measurement uncertainties are likely to destabilize it from this point.

The distance between two particular unit quaternion orientations can be calculated with

$$d(\mathbf{q}, \mathbf{p}) = 1 - \mathbf{q}^T \mathbf{p} \tag{5}$$

The value of this function is between 0 and 2. If the $\mathbf{q}$ is set as the current attitude and $\mathbf{p} \in \{\mathbf{p}_g, -\mathbf{p}_g\}$ is the goal attitude, then the goal attitude with the shorter path will have smaller value. That is $d(\mathbf{q}, \mathbf{p}_q) < d(\mathbf{q}, -\mathbf{p}_q)$ if $\mathbf{p}_g$ will guide the system to the gaol attitude through the shorter path. Now the function for selecting the appropriate unit quaternion can be defined as

$$\Phi(\mathbf{q}, \mathbf{p}) = \underset{\mathbf{p} \in \{\mathbf{p}_g, -\mathbf{p}_g\}}{\arg\min}\ d(\mathbf{q}, \mathbf{p}) \tag{6}$$

3.3 Satellite model

In order to understand how attitude control system affects the orientation of the satellite, it is useful to know its model. Through the model, a satellite behavior can be inspected with different control methods applied.

3.3.1 Kinematics

In (Wertz, 1978) there is a derivation of kinematic equations with quaternion parameters.

$$\dot{\mathbf{q}}_{ob} = \frac{1}{2} \Omega_{ob} \mathbf{q}_{ob}$$

$$\Omega_{ob} = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix} \tag{7}$$

The angular velocities ω_x, ω_y and ω_z in matrix Ω_{ob} are velocities about the axes of the $\mathcal{F}_b$ in respect to the $\mathcal{F}_o$, that is $\omega^{\mathbf{b}}_{\mathbf{ob}}$. Expression (Sidi, 1997)

$$\omega^{\mathbf{b}}_{\mathbf{ib}} = \omega^{\mathbf{b}}_{\mathbf{io}} + \omega^{\mathbf{b}}_{\mathbf{ob}} \tag{8}$$

states that the angular velocity of the body reference frame in regard to the inertial frame can be decomposed to the angular velocities of body in respect to the orbit frame and angular velocity of orbit frame in respect to the inertial frame. From this $\omega^{\mathbf{b}}_{\mathbf{ob}}$ can be expressed as

$$\omega^{\mathbf{b}}_{\mathbf{ob}} = \omega^{\mathbf{b}}_{\mathbf{ib}} - R^b_o \omega^{\mathbf{o}}_{\mathbf{io}} \tag{9}$$

where R^b_o is rotation matrix from orbit to body reference frame. If circular orbit is assumed, it is easy to determine from orbital dynamics that

$$\omega^{\mathbf{o}}_{\mathbf{io}} = \begin{bmatrix} 0 \\ -\omega_0 \\ 0 \end{bmatrix} \tag{10}$$

Here ω_0 is satellite's orbit angular velocity (Sidi, 1997).

Rotation matrix R^b_o from orbit to body reference frame is computed from unit quaternions as (Sidi, 1997)

$$R^b_o = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{11}$$

3.3.2 Dynamic model

Body of the satellite will be considered rigid, thus the Euler's moment equation (Sidi, 1997) will be used:

$$\mathbf{M} = \dot{h}^b_i = \dot{h}^b_b + \omega^{\mathbf{b}}_{\mathbf{ib}} \times h^b_b \tag{12}$$

In the equation, all vectors are represented in $\mathcal{F}_b$. Here $\dot{h}^b_i$ is the change of overall angular momentum of the body which is equal to the externally applied torques. Change of angular momentum as seen in $\mathcal{F}_b$ is $\dot{h}^b_b$. Last term is contribution to the change by the rotation of the frame $\mathcal{F}_b$.

Total change of angular momentum of the body can be represented as a sum of torques from the actuators and disturbances, $\mathbf{T} = \mathbf{T}_c + \mathbf{T}_d$, respectively. Taking this in the Euler's moment equation and the formula for the angular momentum $h = I\omega$, following equation is produced

$$\mathbf{T} = I\dot{\omega}^{\mathbf{b}}_{\mathbf{ib}} + \omega^{\mathbf{b}}_{\mathbf{ib}} \times I\omega^{\mathbf{b}}_{\mathbf{ib}} \tag{13}$$

I is moment of inertia matrix of the rigid body. The model is simplified when the body reference frame $\mathcal{F}_b$ is used, as the inertia matrix becomes diagonal. Dynamic model in the form of the first order ordinary differential equations is given after rearrangement of the terms and separation by the reference frame axes.

$$\begin{aligned}\dot{\omega}_x &= \frac{\omega_y\omega_z(I_y - I_z) + T_x}{I_x} \\ \dot{\omega}_y &= \frac{\omega_x\omega_z(I_z - I_x) + T_y}{I_y} \\ \dot{\omega}_z &= \frac{\omega_x\omega_y(I_x - I_y) + T_z}{I_z}\end{aligned} \tag{14}$$

Those equations show the interdependency between three principal angular velocities due to the moments of inertia and their reaction to external torques. The angular velocities are the dynamic part of the system's state space.

3.3.3 Gravity gradient

Earth's gravity force weakens with the distance from the surface, and different satellite parts are affected with different gravitational forces. This effect can create torques influencing the satellite's body orientation. Equation for the gravity gradient torque is (Wisniewski, 1996)

$$\begin{aligned}\mathbf{T}_g &= 3\omega_0^2(\mathbf{z}_o^b \times I\mathbf{z}_o^b) \\ \omega_0^2 &= \frac{\mu}{R_e^3}\end{aligned} \tag{15}$$

where μ is the Earth's gravitational constant, R_e is distance from the center of the Earth, z_o^b is a unit vector toward the zenith, the z axis of $\mathcal{F}_o$ and I is the inertial matrix of the satellites body.

Written with the unit quaternion that describes the rotation from orbit to body reference frame, gravity gradient equation becomes

$$\mathbf{T}_g = 6\omega_0^2 \begin{bmatrix}(Iz - Iy)(q_2q_3 + q_1q_4)(-q_1^2 - q_2^2 + q_3^2 + q_4^2) \\ (Ix - Iz)(q_1q_3 - q_2q_4)(-q_1^2 - q_2^2 + q_3^2 + q_4^2) \\ (Iy - Ix)(q_2q_3 + q_1q_4)(q_1q_3 - q_2q_4)\end{bmatrix} \tag{16}$$

3.3.4 Atmospheric drag

Lower orbits of the Earth are susceptible to the effect of atmospheric drag as the atmospheric density and satellite's velocity increase at lower altitudes (Fortescue et al., 2011). Also important factors are the ram area and surface material drag coefficient of the satellite's body parts. The forces acting on the satellite's parts can be divided in to the drag forces along the velocity vector and lift forces perpendicular to it, though the lift forces are much smaller and can be neglected. Equation for these forces is

$$\mathbf{F}_d = -\frac{1}{2}\rho SC\|\mathbf{V}_r\|^2\hat{\mathbf{V}}_r \tag{17}$$

where ρ is the atmospheric density, S is ram area of satellite, C is the drag coefficient of the surface material and $\mathbf{V}_r$ is satellites velocity. $\hat{\mathbf{V}}_r$ is a unit vector in the direction of the satellite's velocity. Then, the torque generated by the forces is

$$\mathbf{T}_d = \mathbf{r}_{cmp} \times \mathbf{F}_d \tag{18}$$

where $\mathbf{r}_{cmp}$ is a vector from center of mass to the center of pressure. As the $\hat{\mathbf{V}}_r$ is the first column of the R_o^b DCM, torque $\mathbf{T}_d$ can be rewritten as

$$\mathbf{T}_d = -\|\mathbf{F}_d\| \begin{bmatrix} 2(q_1q_3 + q_2q_4)r_{cmp_y} - 2(q_1q_2 - q_3q_4)r_{cmp_z} \\ (q_1^2 - q_2^2 - q_3^2 + q_4^2)r_{cmp_z} - 2(q_1q_3 + q_2q_4)r_{cmp_x} \\ 2(q_1q_2 - q_3q_4)r_{cmp_x} - (q_1^2 - q_2^2 - q_3^2 + q_4^2)r_{cmp_y} \end{bmatrix} \tag{19}$$

It is difficult to precisely account for the atmospheric drag disturbance, current models are only modestly precise (Wertz, 1978). As stated in the mission description, the region of lower thermosphere where the satellite's orbit will be is the least researched and the mission's purpose is to get data for improving existing models. Atmospheric density depends on multiple factors and is quite variable. It varies along the altitude, latitude and with intensity of sun radiation and its activity. The ram area is dependent on the orientation of the satellite's surfaces, center of pressure changes with the area of exposed surface, and drag coefficient varies with the molecules flow regime (Fortescue et al., 2011).

Drag torques can be reduced by careful structural design which minimizes the vector from center of mass to the possible centers of pressure. Also the small size of the Aalto-2 satellite weakens this torque, as the ram area is small.

On the other hand, if the satellite's body is aerodynamically designed, the atmospheric drag torque can be used for passive attitude stabilization. Because the atmospheric drag will be the most significant disturbance torque, it is investigated if it can be used to improve the attitude stability by adjusting the satellite's body elements.

3.3.5 Solar pressure

Solar pressure is generated by the electromagnetic radiation from the Sun that illuminates the satellites surface (Fortescue et al., 2011). Momentum from the radiation is exchanged with the satellites body. Equation of the solar pressure is

$$P = \frac{F_s}{c} \tag{20}$$

with F_s being the solar energy flux with the approximate value of $1400 W/m^2$ near Earth and c is the speed of light. Force created with this pressure is given with the equation

$$\mathbf{F}_{sp} = sAPr_f\hat{\mathbf{r}}_{sp} \tag{21}$$

In this equation the s is a reflectivity factor. It depends on the material and its value vary from 0 for completely transparent to 2 for completely reflective. A is the illuminated surface area, r_f is the ration between the distance of the satellite and

Earth from the Sun, and $\hat{\mathbf{r}}_{sp}$ is the unit vector pointing from the Sun to the center of solar pressure on the satellite's body surface.

The torque generated with this force is

$$\mathbf{T}_{sp} = \mathbf{r}_{cmp} \times \mathbf{F}_{sp} \tag{22}$$

where $\mathbf{r}_{cmp}$ is the vector from the center of mass to the center of pressure.

As with the atmospheric drag, the small size of the satellite reduces the effect of this torque and it can be lowered additionally with careful structural design. This torque will be considered negligible during the control of attitude.

3.3.6 Magnetic actuators

The attitude control with the magnetic actuators can only produce torques in a plane normal to the Earth's magnetic field. This effect limits the controllability of the satellite. However, the full three-axis control can be achieved if the magnetic field changes around the satellite periodically, providing different planes for the torque generation over time in $\mathcal{F}_b$.

Periodic change of the magnetic field vector depends on the satellite's orbit. The satellite in the equator orbit will always have the same Earth magnetic field vector in $\mathcal{F}_o$ and would always have limited controllability. As the orbit moves closer to the polar orbit, periodical change gets more rapid.

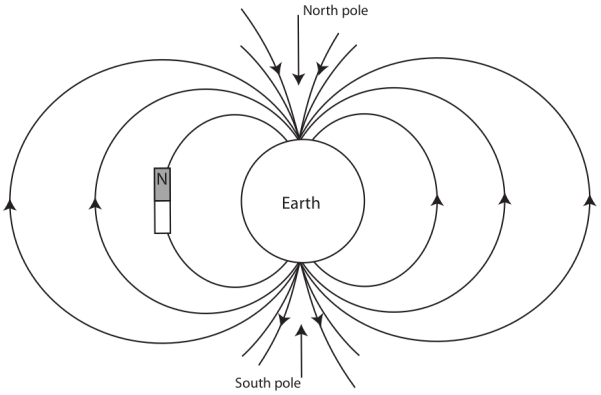

Figure 2: Earth's dipole model, (Tudor, 2011)

Control of the satellite attitude with the magnetic actuators is possible due to the torques that are produced by the interaction of the magnetic field they generate and the magnetic field of the Earth. The equation for torque is

$$\mathbf{T}_c = \mathbf{m} \times \mathbf{B} \tag{23}$$

In this equation, magnetic dipole moment from magnetic actuators is denoted with $\mathbf{m}$, and Earth's magnetic field with $\mathbf{B}$. For a three axis control, magnetic coils can be oriented in a way for the torque vectors to align with the axes of the body reference frame.

3.4 Stability

While studying the control methods and their laws it is important to be able to determine whether the method makes the system stable. System is considered stable when its state stays within some boundary. This stability type is also referred to as stability in the sense of Lyapunov. A more stricter type of stability, asymptotic stability, have a property that the system converges its state to an equilibrium point.

Stability in the sense of Lyapunov might be expressed formally as follows (Slotine and Li, 1991):
Equilibrium state $\mathbf{x} = 0$ *is said to be stable if, for any* $R > 0$*, there exists* $r > 0$*, such that if* $\|\mathbf{x}(0)\| < r$*, then* $\|\mathbf{x}(t)\| < R$ *for some* $t \geq 0$*. Otherwise the system is unstable.*

And definition of the asymptotic stability is:
An equilibrium point is asymptotically stable if it is stable in the sense of Lyapunov, and if in addition there exists some $r > 0$ *such that* $\|\mathbf{x}(0)\| < r$ *implies that* $\mathbf{x}(t) \to 0$ *as* $t \to \infty$*.*

If the stability of the system is not dependent on the time, that is the definitions hold for all $t \geq 0$, the stability of system is also uniform. There, additionally can be made a distinction between the local and global asymptotic stability. System is globally stable if it is stable for any starting condition.

3.4.1 Direct method of Lyapunov

The Direct method of Lyapunov is used when determining the stability of non-linear systems. If it is possible to construct a Lyapunov function $V(\mathbf{x})$ for a system, then the system is stable in the sense of the Lyapunov. This function should represent the energy of the system, it needs to be continuously differentiable, and to have following properties

$$\begin{aligned} V(\mathbf{x}) = 0 \text{ when } \mathbf{x} = 0 \\ V(\mathbf{x}) > 0 \text{ when } \mathbf{x} \neq 0 \\ \dot{V}(\mathbf{x}) \leq 0 \end{aligned} \tag{24}$$

And for asymptotic stability next properties need to hold

$$\begin{aligned} V(\mathbf{x}) = 0 \text{ and } \dot{V}(\mathbf{x}) = 0 \text{ when } \mathbf{x} = 0 \\ V(\mathbf{x}) > 0 \text{ and } \dot{V}(\mathbf{x}) < 0 \text{ when } \mathbf{x} \neq 0 \end{aligned} \tag{25}$$

Requirements for the stability in this method can be interpreted as the need to dissipate the energy of system as long as the state is away from the equilibrium point.

For the analysis of periodic systems in some cases this method is sufficient, but there are also few variants of this method. One of them introduces the periodic Lyapunov function and relaxes the requirements (Böhm et al., 2012).

3.4.2 Satellite stability

Total energy of the satellite is sum of kinetic and potential energies in inertial reference frame $\mathcal{F}_i$. However, to check the attitude stability in orbit it is enough to consider only the energy in orbit frame $\mathcal{F}_o$, so the kinetic energy from satellite's movement around the Earth will be disregarded. The remaining kinematic energy from angular velocity is

$$E_k = \frac{1}{2}\omega_{\mathbf{ob}}^{\mathbf{b}\ T} I \omega_{\mathbf{ob}}^{\mathbf{b}} \tag{26}$$

Potential energy of the satellite is a sum of the energies associated with the gravity gradient, atmospheric drag and satellite's revolution around the Earth (Sarychev et al., 2007), and it is given as

$$E_p = \frac{3}{2}\omega_0^2(\mathbf{z}_o^{b\,T} I \mathbf{z}_o^b - I_x) + \frac{1}{2}\omega_0^2(I_z - \mathbf{x}_o^{b\,T} I \mathbf{x}_o^b) + \|\mathbf{F}_d\|(\|\mathbf{r}_{cmp} - R_o^b \mathbf{r}_{cmp}\|) \tag{27}$$

where $\mathbf{z}_o^b$ is a z axis and $\mathbf{x}_o^b$ is x axis of orbit reference frame $\mathcal{F}_o$ as seen in body reference frame $\mathcal{F}_b$. I_x and I_z are minor and major principal inertia axes.

Total satellite's energy in orbit frame will be used as the Lyapunov function

$$V(\omega_{\mathbf{ob}}^{\mathbf{b}}, \mathbf{q}_{ob}) = E_k(\omega_{\mathbf{ob}}^{\mathbf{b}}) + E_p(\mathbf{q}_{ob}) \tag{28}$$

The kinetic part of the energy equation have a quadratic multiplier of the angular velocity, which is the passed in argument to the Lyapunov function, and it is equal to zero in the subspace of the state space with the zero angular velocity. However, the potential energy can never be zero. Its first two terms, gravity gradient and Earth revolution potential energies, have their minimum values of zero when the satellite's body x axis is aligned with the orbit's z axis, and they are at their maximums when the axes are perpendicular. The potential energy part coming from the atmospheric drag is at its minimum when the satellite's velocity vector is aligned with the vector from the center of mass to the center of pressure, and is at the maximum when they are perpendicular. As the center of pressure vector is nearly aligned with the body's x axis, it is clear that when the atmospheric drag potential energy is at the minimum, the gravity gradient and Earth revolution potential energies are at the maximum, and vice versa.

This violates the first requirement of the Direct method of Lyapunov (24), which means that the satellite is inherently unstable at the altitudes with significant atmospheric drag disturbance. If uncontrolled and with some amount of kinteic energy, the satellite's attitude will oscillate around the point of minimum energy. Location of this point is dependent on the moment of inertia matrix and the strength of the atmospheric drag disturbance. The figure 3 shows the polar plot of potential energy strength when the satellite is rotated around its y axis for three different altitudes, 400, 300 and 200 km. The innermost contour shows the energy at the altitude of $400km$, and the outermost contour shows the energy at $200km$. At higher altitudes gravity gradient disturbance dominates and satellite's attitude will tend to align with the zenith vector. As the altitude gets lower, atmospheric disturbance

increases and shifts the point of minimum energy toward the velocity vector. However, slope around the minimum points indicate the strength of the oscillations of attitude around them, so they are expected to be higher at the lower altitudes.

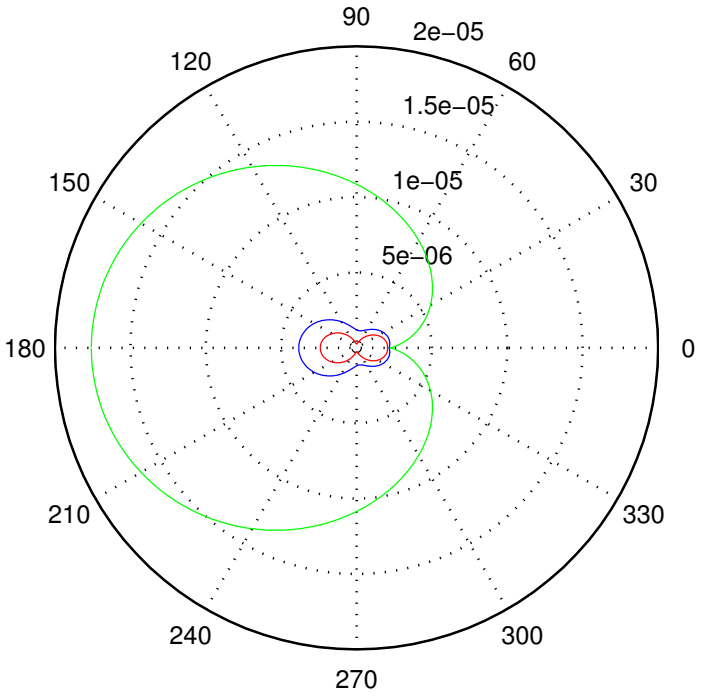

Figure 3: Potential energy around y axis

Active control is needed in order to keep the satellite in the desired attitude, and from the figure 3 it can be seen that the control should be able to withstand a big change in the environemnt. This project investigates the use of magnetic actuators with passive atmospheric stabilization, and examines if the setup is sufficient for the stabilizing of the attitude and the performance of several different control methods.

4 Attitude Control

The goal of the attitude control is to navigate the satellite's body orientation to a desired attitude and keep it stable in that reference orientation point. The reference point for the Aalto-2 satellite is designated by the velocity vector, the x axis of body reference frame needs to be aligned with it. When the satellite is in the circular orbit, its velocity vector coincides with the x axis of orbit reference frame $\mathcal{F}_o$. As the mission's orbit is nearly circular, orbit reference frame will be considered as the goal frame. That is, the satellite have the correct attitude when the body and orbit reference frames are aligned. The difference between the desired attitude and satellite's current attitude is expressed with unit quaternion and it is calculated with the equation (3). It is denoted with $\mathbf{q}_{ob}$. The attitude control goal can be expressed as:

$$\mathbf{q}_{ob} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{29}$$

The desired attitude unit quaternion can be found from the satellite's current position in orbit and velocity vectors in inertial reference frame, $\mathcal{F}_i$. First DCM is constructed taking the velocity vector as x axis, y axis is a cross product of x axis and negative position vector, and z axis is found from the cross product of x and y axes. Lengths of these vectors also need to be normalized to unit lengths. This DCM can now be converted to unit quaternion.

Attitude control can be divided to two different modes, reference point tracking and detumbling. Reference point tracking mode is concerned with keeping the orientation difference (29) as small as possible, while the detumbling mode is used to reduce high angular velocities.

Essential counterpart of the Attitude Control System is the Attitude Determination System (ADS). The ADS provides current values of the attitude and angular velocity of satellite, relaying on the satellite and environment models and sensor measurements. The two of the systems cooperate in controling of the attitude, thus both of the systems influence the overall pointing precision. As the ADS is developed simultaniuously with the ACS, this project does not include its effects on the attitude precision, still the upper limit of the ACS efficiency can be studied. Anyhow, final control precision, before the mission start needs to be evaluated with both systems active.

Multiple control methods will be presented and their working principles will be explained in the upcoming subsections.

4.1 Passive atmospheric drag stabilization

Since the atmospheric drag disturbance is expected to be the dominant disturbance, it can be beneficial to use it for improving of the satellite's attitude stability. It is also convenient to use it, as the desired attitude is tied to the velocity vector, and from the atmospheric drag force equation (17) it can be seen that the atmospheric

drag force vector coincides with it. From this it can be concluded that the torque generated with this force (18) will act to rotate the satellite toward its desired attitude as long as the vector from the center of mass to the center of pressure, $\mathbf{r}_{cmp}$, is positioned along the satellite's x axis, in the direction from the ram to the wake of the satellite. Torque will rotate the satellite's body around the center of mass to align the body's x axis with the velocity vector, equally, the vector from center of mass to the center of pressure will tend to align with the atmospheric drag force vector. This is shown on the figure 4. As there is little energy dissipation, this torque will mostly generate oscillation around the desired attitude. The oscillation can be, to some extent, damped by the active control methods.

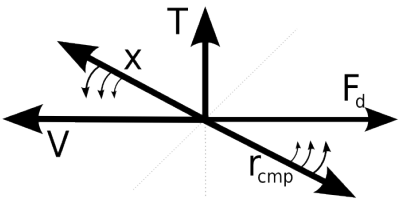

Figure 4: Atmospheric drag torque

To set the vector $\mathbf{r}_{cmp}$, the center of mass can be adjusted by the mass distribution of the satellite within $2cm$ about the geometrical center of the body. This is the maximal allowed displacement defined by the CubeSat standard (Munakata et al., 2009). The center of pressure is variable property which depends on the exposed surface area to the atmospheric drag. It is hard to predict the exact location of this property. However, for the symmetrical bodies the geometrical center point can be a good approximation. After the satellite's deployment in orbit, it will extend four antennas, which are positioned at the satellite's rear end, with the angle of 135° relative to the body's x axis, as shown in the figure 5. They should also affect the torque acting on the satellite, improving the tendency toward the desired atitude.

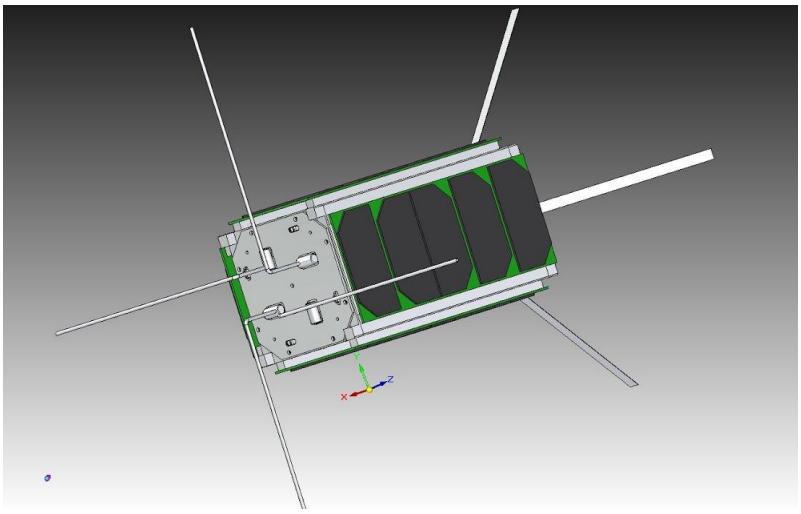

Figure 5: Aalto-2 satellite body

4.2 Detumbling control

It is very probable that the satellite will have a high angular velocity upon its deployment in the orbit. Reference point tracking mode control methods are inefficient or incapable of reaching the reference point under these rapid rotations. Thus, angular velocity should be reduced before the reference point tracking mode is activated. Detumbling control is designed to be effective and robust in slowing down of the satellite's rotations.

The most simple and often used method for detumbling of satellite with magnetic actuators is $\dot{\mathbf{B}}$ (Silani and Lovera, 2005). This method is based on dissipation of the energy of satellite's body. The control law of this method is

$$\mathbf{m} = -K\dot{\mathbf{B}}(t) \tag{30}$$

with K being a positive definite gain matrix and the differentiated magnetic field

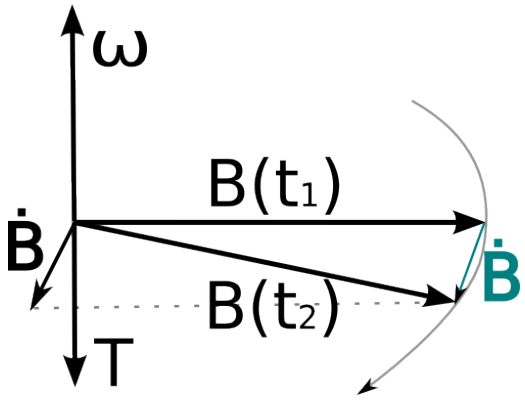

Figure 6: $\dot{\mathbf{B}}$ vectors

vector is expressed in the body reference frame $\mathcal{F}_b$. Vectors involved with this control can be seen on Figure 6. Time derivative of the magnetic field is perpendicular to the vector of rotation as long as the body angular velocity is sufficiently higher than the orbit angular velocity. The derivative is perpendicular to the magnetic field vector as well. This gives the following approximation

$$\dot{\mathbf{B}} \approx \mathbf{B} \times \omega_{\mathbf{ob}}^{\mathbf{b}} \tag{31}$$

Negating the derivative vector and multiplying with the gain matrix produces the control law. When the control law is applied to the actuators, it creates the magnetic dipole moment that through the equation (23) generates the torque which in general opposes the angular velocity of the body, that is it reduces the change of magnetic field as seen in body's reference frame to zero.

Satellite controlled with this law will dissipate its energy in a way that the remaining rotation will follow the orientation of the vector of the magnetic field. Although with this law the satellite remains with some angular velocity, it is low enough for the most control methods to carry on with their control successfully. Time derivative of the Earth's magnetic field can be easily measured with on-board sensors.

4.3 PD control

In the field of automation and control, most prominent controllers are of a proportional-integral-derivative class. They have gained their popularity due to the simplicity and success in controlling of a large variety of plant types.

Use of the PD controller for attitude control has been shown successful in Wen and Kreutz-Delgado (1991) for completely controllable rigid bodies. This is the case with the reaction wheels or thrusters, but with magnetic torquers there is a lack of controllability about one axis at any time. However, Silani and Lovera (2005) argue that in orbits with periodic change of magnetic field it is possible to control the satellite over time and reach closed-loop stability, if closed-loop dynamics are sufficiently slow. Stability is usually checked aposteriori through the Floquet theory.

Three control laws are presented in Wen and Kreutz-Delgado (1991). Their outputs should not be directly applied to the actuators. Instead the output should be projected on the plane perpendicular to the vector of Earth's magnetic field, as to minimize the energy consumption. Also, the output should be converted to the magnetic dipole moment.

The conversion that takes care of both of this requirements is given with the formula (Silani and Lovera, 2005)

$$\mathbf{m} = \frac{1}{\|\mathbf{B}\|^2}\mathbf{B} \times \mathbf{T}_c \tag{32}$$

It also projects the torque vector to the plane normal to the magnetic field vector. With projection, only the usable part of the magnetic dipole moments is generated, lowering energy consumption. This projection is shown in Figure 7.

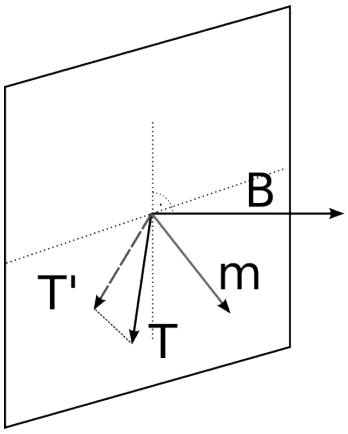

Figure 7: Projection of torque on plane normal to Earth's magnetic field vectors

The model independent control law has been adapted in Tudor (2011), and its form is of a basic PD.

$$\mathbf{T}_c = -K_p\overline{\mathbf{q}}_{ob} - K_d\omega_{\mathbf{ob}}^{\mathbf{b}} \tag{33}$$

K_p and K_d are positive coefficients. The law is found through the Lyapunov function for a fully controllable system model. Since the magnetic actuation is not fully

controllable, additional testing needs to be done, possibly via Floquet theory. Tudor (2011) states that the control law is locally uniformly asymptotically stable.

Second control law in Wen and Kreutz-Delgado (1991) is model dependent law. It improves the control efficiency by including information about the model in the law.

$$\mathbf{T}_c = -K_p \overline{\mathbf{q}}_{ob} - K_d \omega_{\mathbf{ob}}^{\mathbf{b}} + I\dot{\omega}_{\mathbf{ib}}^{\mathbf{b}} + \omega_{\mathbf{ob}}^{\mathbf{b}} \times I\omega_{\mathbf{ob}}^{\mathbf{b}} \tag{34}$$

Inertia matrix I brings the information about the satellite in the equation.

Last control law presented is for adaptive control. It builds on the model dependent control law by allowing the inertia matrix to be updated by a better estimate. New estimate can be made by a gradient method.

Since the size of the Aalto-2 satellite's body is light and its inertia matrix components are small, model dependent terms are negligible. Only model independent variant is considered for this project.

4.4 Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) belongs to a class of optimal controllers. With optimal approaches it is possible to introduce optimization of the control with the respect to a selected parameter. For instance, attitude control could be optimized to reach its goal orientation with minimum time or energy used. Optimization condition is defined through the cost functional and its value needs to be minimized under the control that guides the system to a wanted state. Within the class of optimal control there are multiple approaches. Most direct method would be the use of calculus of variations, although it is rarely used due to its limitation to specific problems. Conversely, the Linear Quadratic Regulator have been implemented for many control problems, as it is applicable on various plants and its theory is well understood. Though this method have a limitation to be used only on linear systems that are fully controllable, model linearization can be used to allow its application to a non-linear systems too.

A Linear Quadrature Regulator with infinite horizon (Speyer and Jacobson, 2010) have been selected. Infinite horizon variant of LQR does not put any costs on the time needed to reach the reference point. As the dynamics of the system need to evolve slowly because of the controllability limitations linked with the magnetic actuation, infinite horizon is convenient for the problem at hand.

For the linear system $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$, with A being a system matrix, B being a control matrix and given starting state $\mathbf{x}_0$, the cost functional for infinite horizon variant of LQR is defined as

$$J = \frac{1}{2} \int_{t_0}^{\infty} [\mathbf{x}^T(t) Q \mathbf{x}(t) + \mathbf{u}^T(t) R \mathbf{u}(t)] dt \tag{35}$$

Q is a real symmetric positive semi-definite and R is real symmetric positive definite matrix. Controller can be adjusted by setting the Q and R matrices. Q matrix defines the cost of the state error, while R defines the cost of the control effort.

Control law can be found to be

$$\begin{aligned} \mathbf{u}(t) &= K(t)\mathbf{x}(t) \\ K(t) &= -R^{-1}B^T(t)L(t) \end{aligned} \tag{36}$$

where $L(t)$ is a solution to the Algebraic Matrix Riccati Equation (AMRE)

$$L(t)A + A^T L(t) - L(t)B(t)R^{-1}B^T(t)L(t) + Q = 0 \tag{37}$$

The solution of the AMRE is also periodic due to the periodicity of matrix $B(t)$ when the conversion of the control signal to the magnetic dipole (32) is applied.

4.4.1 Model linearization

In order to apply the LQR to a satellite attitude control the model of the satellite needs to be linearized. Linearization of dynamic equations can be carried out by approximation of rotations with small angles around a selected attitude. The attitude around which the linearization will be done is when the $\mathcal{F}_b$ coincides with $\mathcal{F}_o$. This approximates unit quaternion of the difference between the desired and current attitude to

$$\mathbf{q}_{ob} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \sim \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{38}$$

Linearized gravity gradient torque Applying the approximated unit quaternion (38) to the gravity gradient torque (16) gives

$$\mathbf{T}_g = 6\omega_0^2 \begin{bmatrix} -(Iy - Iz)q_1 \\ (Iz - Ix)q_2 \\ 0 \end{bmatrix} \tag{39}$$

Linearized atmospheric drag torque Same as for the gravity gradient, using the small value approximation (38) to the atmospheric drag torque (19) and noticing that the $\mathbf{r}_{cmp}$ vector is parallel with x axis of $\mathcal{F}_b$ and with opposite direction ($\mathbf{r}_{cmp} = [-\|\mathbf{r}_{cmp}\|\ 0\ 0]^T$), linearized torque is produced

$$\mathbf{T}_d = -2\|\mathbf{r}_{cmp}\|\|\mathbf{F}_d\| \begin{bmatrix} 0 \\ q_2 \\ q_3 \end{bmatrix} \tag{40}$$

Values for the vectors $\mathbf{F}_d$ and $\mathbf{r}_{cmp}$ are variable and difficult to predict. For that reason they will be approximated, according to the current orbit parameters (ρ and V_r). Center of pressure $\mathbf{r}_{cmp}$ and exposed surface area S will have fixed values, though in reality they are also variable.

Linearized kinematic equations For linearization of the kinematics equations, it is important to linearize the R_o^b direction cosine matrix (11). Applying unit quaternion approximation (38) to it gives

$$R_o^b = \begin{bmatrix} 1 & 2q_3 & -2q_2 \\ -2q_3 & 1 & 2q_1 \\ 2q_2 & -2q_1 & 1 \end{bmatrix} \tag{41}$$

Then by applying the same approximation (38) to non-linear kinematic equations (7), linearized kinematic equations are produced

$$\dot{\mathbf{q}}_{ob} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\omega_{obx}^b \\ \frac{1}{2}\omega_{oby}^b \\ \frac{1}{2}\omega_{obz}^b \\ 0 \end{bmatrix} \tag{42}$$

Linearized dynamic equations To linearize the dynamic equations, kinematic approximations have to be applied to them. First $\omega_{\mathbf{io}}^{\mathbf{b}}$ will be calculated by taking orbit angular velocity in $\mathcal{F}_o$ (10) and applying linearized rotation matrix (41) to it

$$\omega_{\mathbf{io}}^{\mathbf{b}} = \begin{bmatrix} -2q_3\omega_0 \\ -\omega_0 \\ 2q_1\omega_0 \end{bmatrix} \tag{43}$$

Then with equations (8) and (43) $\omega_{\mathbf{ib}}^{\mathbf{b}}$ can be calculated as

$$\omega_{\mathbf{ib}}^{\mathbf{b}} = \omega_{\mathbf{ob}}^{\mathbf{b}} + \begin{bmatrix} -2q_3\omega_0 \\ -\omega_0 \\ 2q_1\omega_0 \end{bmatrix} \tag{44}$$

By differentiating angular velocity of satellite's body $\omega_{\mathbf{ib}}^{\mathbf{b}}$, it is obtained

$$\dot{\omega}_{\mathbf{ib}}^{\mathbf{b}} = \dot{\omega}_{\mathbf{ob}}^{\mathbf{b}} + \begin{bmatrix} -2\dot{q}_3\omega_0 \\ 0 \\ 2\dot{q}_1\omega_0 \end{bmatrix} \tag{45}$$

Finally inserting equations (39), (40), (44) and (45) into dynamic equations (14) and eliminating very small products, linearized dynamic model is produced

$$\begin{aligned} \dot{\omega}_{\mathbf{ob_x}}^{\mathbf{b}} &= \omega_0(1-k_x)\omega_{\mathbf{ob_z}}^{\mathbf{b}} - 8k_x\omega_0^2 q_1 + \frac{T_{cx}}{I_x} \\ \dot{\omega}_{\mathbf{ob_y}}^{\mathbf{b}} &= (6k_y\omega_0^2 - \frac{2\|\mathbf{r}_{cmp}\|\|\mathbf{F}_d\|}{I_y})q_2 + \frac{T_{cy}}{I_y} \\ \dot{\omega}_{\mathbf{ob_z}}^{\mathbf{b}} &= -\omega_0(1+k_z)\omega_{\mathbf{ob_x}}^{\mathbf{b}} + (2k_z\omega_0^2 - \frac{2\|\mathbf{r}_{cmp}\|\|\mathbf{F}_d\|}{I_z})q_3 + \frac{T_{cz}}{I_z} \end{aligned} \tag{46}$$

where

$$\begin{aligned} k_x &= \frac{(I_y - I_z)}{I_x} \\ k_y &= \frac{(I_z - I_x)}{I_y} \\ k_z &= \frac{(I_x - I_y)}{I_z} \end{aligned} \tag{47}$$

State space representation After the linearized equations of the model are obtained, the state space vector can be defined as

$$\mathbf{x} = \begin{bmatrix} \overline{\mathbf{q}}_{ob} \\ \omega^{\mathbf{b}}_{\mathbf{ob}} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \omega^{\mathbf{b}}_{\mathbf{ob_x}} \\ \omega^{\mathbf{b}}_{\mathbf{ob_y}} \\ \omega^{\mathbf{b}}_{\mathbf{ob_z}} \end{bmatrix} \tag{48}$$

Now linearized state space model looks like

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{m} \tag{49}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -8k_x\omega_0^2 & 0 & 0 & 0 & 0 & \omega_0(1-k_x) \\ 0 & 6k_y\omega_0^2 - \frac{T_d}{I_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & -2k_z\omega_0^2 - \frac{T_d}{I_z} & -\omega_0(1+k_z) & 0 & 0 \end{bmatrix}, \tag{50}$$

$$T_d = 2\|\mathbf{r}_{cmp}\|\|\mathbf{F}_d\|$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{B_z}{I_x} & \frac{-B_y}{I_x} \\ \frac{-B_z}{I_y} & 0 & \frac{B_x}{I_y} \\ \frac{B_y}{I_z} & \frac{-B_x}{I_z} & 0 \end{bmatrix} \tag{51}$$

4.5 State Dependent Riccati Equation

As the dynamic model of the satellite is non-linear, the use of the LQR is efficient only near the linearization point. Linearization brings with it imprecisions while the system state is far from the linearization point. This ultimately limits the use of those approaches to local attitude control. A way to evade this limitation is introduced with the State Dependent Riccati Equation (SDRE) approach. This

method allows the use of the non-linear model with the LQR theory (Abdelrahman et al., 2011).

SDRE transforms the nonlinear model of the system into a quasi-linear model, or alternately called State Dependent Coefficient (SDC) form.

$$\dot{\mathbf{x}}(t) = A(\mathbf{x})\mathbf{x}(t) + B(\mathbf{x})\mathbf{u}(t) \tag{52}$$

Matrices $A(\mathbf{x})$ and $B(\mathbf{x})$ are dependent on the current system state. Parametrization of the matrices is not unique, there are multiple ways to express them. Now, it is possible with this model to apply the approach of the infinite horizon LQR controller. That will produce a different Algebraic Matrix Riccati equation, and it will be dependent on the system state.

$$L(\mathbf{x})A(\mathbf{x}) + A(\mathbf{x})^T L(\mathbf{x}) - L(\mathbf{x})B(\mathbf{x})R^{-1}B(\mathbf{x})^T L(\mathbf{x}) + Q = 0 \tag{53}$$

Solution of this SDRE can be used to get the gain matrix K

$$K(\mathbf{x}) = -R^{-1}B(\mathbf{x})^T L(\mathbf{x}) \tag{54}$$

It is obvious that the gain matrix is also state dependent. Because of this it is necessary to recalculate the gain matrix $K(\mathbf{x})$ every step of the control.

4.5.1 SDC parametrization of satellite model

One parametrization of a satellite model with magnetic actuators can be found in Abdelrahman et al. (2011). This SDC will be adopted and adapted for the Aalto-2 satellite.

The system matrix $A(x)$ will be partitioned in four submatrices. Dimensions of theses parts are 3x3, and the partition is shown below

$$A(x) = \begin{bmatrix} A_{11}(\omega^{\mathbf{b}}_{\mathbf{ob}}) & A_{12}(\mathbf{q}_{ob}) \\ A_{21}(\mathbf{q}_{ob}) & A_{22}(\omega^{\mathbf{b}}_{\mathbf{ob}}) \end{bmatrix} \tag{55}$$

Submatrices $A_{11}(x)$ and $A_{12}(x)$ show the influence of kinematic and dynamic parts of state space on the kinematic equations, respectively. From kinematic equations (7) these submatrices can be formed as

$$A_{11}(\omega^{\mathbf{b}}_{\mathbf{ob}}) = \frac{1}{2}\begin{bmatrix} 0 & \omega^b_{ob_z} & -\omega^b_{ob_y} \\ -\omega^b_{ob_z} & 0 & \omega^b_{ob_x} \\ \omega^b_{ob_y} & -\omega^b_{ob_x} & 0 \end{bmatrix} = -\frac{1}{2}skew(\omega^{\mathbf{b}}_{\mathbf{ob}}) \tag{56}$$

$$A_{12}(\mathbf{q}_{ob}) = \frac{1}{2}\begin{bmatrix} q_4 & 0 & 0 \\ 0 & q_4 & 0 \\ 0 & 0 & q_4 \end{bmatrix} \tag{57}$$

In order to parametrize the dynamic equations (14) of the model in the submatrices $A_{21}(x)$ and $A_{22}(x)$, the equation (9) is inserted in them. Then the resulting equation is combined into the differentiated equation (9). The torque term is separated in the gravity gradient, atmospheric drag and magnetic dipole parts. Finally

equations (16), (19) and (23) are included, assuming that the $\mathbf{r}_{cmp}$ vector is parallel to the body's x axis and with opposite direction. Atmospheric drag torque variables are approximated, same as for the LQR. From this the submatrices are formed.

$$\begin{aligned} A_{21}(\mathbf{q}_{ob}) = -\omega_0^2 I^{-1} skew(R^b_{o2}) I \begin{bmatrix} 2q_2 & 0 & 2q_4 \\ q_1(r-2) & q_2 r & q_3(r-2) \\ -2q_4 & 2q_3 & 0 \end{bmatrix} ... \\ +3\omega_0^2 I^{-1} \begin{bmatrix} -2(I_y - I_z)q_4 R^b_{o33} & 0 & -2(I_y - I_z)q_2 R^b_{o33} \\ -2(I_z - I_x)q_3 R^b_{o33} & 2(I_z - I_x)q_4 R^b_{o33} & 0 \\ 0 & 2(I_x - I_y)q_4 R^b_{o32} & -2(I_x - I_y)q_1 R^b_{o32} \end{bmatrix} ... \\ +2\|\mathbf{r}_{cmp}\|\|\mathbf{F}_d\| I^{-1} \begin{bmatrix} 0 & 0 & 0 \\ -q_3 & -q_4 & 0 \\ 0 & q_1 & -q_4 \end{bmatrix} \\ r = \frac{1}{1 - q_4^2} \end{aligned} \tag{58}$$

$$\begin{aligned} A_{22}(\omega^{\mathbf{b}}_{\mathbf{ob}}) = \omega_0 skew(R^b_{ob2}) - I^{-1} skew(\omega^{\mathbf{b}}_{\mathbf{ob}}) I ... \\ -\omega_0 I^{-1} skew(I R^2_{ob2}) + \omega_0 I^{-1} skew(R^b_{ob2}) I \end{aligned} \tag{59}$$

Input matrix B is independent of the system state, though it still depends on the Earth's magnetic field, and is the same as in the linearized model (51)

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{B_z}{I_x} & \frac{-B_y}{I_x} \\ \frac{-B_z}{I_y} & 0 & \frac{B_x}{I_y} \\ \frac{B_y}{I_z} & \frac{-B_x}{I_z} & 0 \end{bmatrix} \tag{60}$$

5 Solving of Algebraic Matrix Riccati Equation

The crucial part of the LQR and SDRE control methods is the solver for the Algebraic Matrix Riccati Equations (37)(53). The Riccati equation solutions provide the dynamic system with the optimal control signals for the well posed problems formulated with the linear-quadratic cost functional (35). The research on this topic have been long and exhaustive. Though, finding a solution for the system of linear-quadratic equations remains complex and computationally intensive problem.

Algebraic Matrix Riccati Equations might have many solutions. However, only one solution can produce stabilizing control signal for a closed loop dynamic system (Katsev, 2006). In order for this solution to exist, which is also symmetric and positive semi-definite, several requirements need to be satisfied (Sima, 1996).

- (A, B) matrix pair needs to be stabilizable.
- (A, D) matrix pair needs to be detectable.
- Q matrix should be symmetric positive semi-definite.
- R matrix should be symmetric positive definite.

The matrix pair (A, B) is considered stabilizable if $rank([\lambda I - AB]) = n$ for all complex eigenvalues λ. And the pair of matrices (A, D) is detectable if and only if the pair (D^T, A^T) is stabilizable. Matrix D is a Cholesky factor of $Q = D^T D$.

Multiple methods have been developed for solving of the Algebraic Matrix Riccati Equations. Generally they can be divided in two groups, the numerically robust and numerically iterative algorithms (Katsev, 2006).

The solution of the AMRE can be guaranteed for numerically robust algorithms in mostly constant computation time. On the other hand, iterative algorithms require a starting guess on which the convergence to the solution depends. With a poor guess the algorithm can converge to a wrong solution. However, computationally wise, iterative methods can outperform the numerically robust algorithms if their starting guess is relatively accurate. In the case of the continuous-time dynamic systems it is assumed that the change of the control signal is also continuous. This can be utilized to reuse the solution from previous iteration as a starting guess. Obtaining the control signal can be efficient in this way. Also the memory requirements of the robust algorithms are usually higher. Still, there remains the need for the good initial guess. Also, in the case of the SDRE control method, depending on the SDC parametrization, the system matrix can be badly formed in some regions of the state space. Until the system reaches a favorable state again, solution of the AMRE can be far from the last proper solution. In those cases it is useful to obtain the solution via a numerically robust algorithm. This approach have been suggested in Menon et al. (2002) and Katsev (2006).

Among the numerically stable algorithms, the Schur decomposition method have been selected as it is considered as the de-facto standard for solving the AMRE. For the iterative method the choice is the Kleinman algorithm, which shows really fast convergence capability.

Although for the project this two groups of algorithms will be considered, it is worthy to mention two other approaches.

When the dynamic system is periodic, the solution of the AMRE is periodic too. This fact can be used to simplify the problem and reduce the calculation requirements. This approach have found a good reception for the magnetically actuated satellites, as the sensed Earth's magnetic field is periodic over the orbit (Wisniewski, 1996), (Lovera et al., 2002), though stability is not guaranteed and have to be additionally checked. This approach is possible only for the LQR control as the SDRE method disturbs the periodicity with its parametrization of the system matrix A. Thus, this approach was rejected.

Recently, artificial neural network methods have been applied to some control problems (Lin and Chen, 2001), (Yucelen et al., 2010). Gradient type recurrent neural networks can produce an approximate solution and are convenient for modern parallel processing structures. This approach can be reduced to the iterative method, although with some requirements fulfilled the convergence is assured independently of the starting guess. That is, this approach can be sufficient, there is no need for the good first guess. However, the algorithms from first two groups were favorable as they are more mature and considerably documented.

5.1 Schur decomposition algorithm

Schur decomposition algorithm provides the solution if the four requirements listed above are fulfilled. This method is largely used and applied in many control solutions due to its robustness and reliability.

The solution is sought by determining the basis of the invariant subspace of an extended matrix associated with the optimal problem (Sima, 1996), (Bunse-gerstner). In the case of the LQR, this extended matrix can be the Hamiltonian matrix H.

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \tag{61}$$

This matrix have n eigenvalues on the left side of the imaginary axis which define the invariant subspace. An orthogonal matrix U, that transforms the H in the Real Schur form, contains this subspace. It can be obtained with the Schur decomposition $H = U^T T U$. Matrix T is an upper diagonal and it have blocks of sizes 1x1 for real and 2x2 for complex eigenvalues along its diagonal. As the Schur decomposition algorithms usually can not provide any specific ordering of the eigenvalues along the diagonal, next step is their reordering in a way that all stable eigenvalues are in the upper part of the matrix diagonal. That is, if the Schur decomposition matrices are defined with following partitions

$$\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}^T H \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \tag{62}$$

Then the submatrix T_{11} should contain all the stable eigenvalues. This step modifies both T and U matrices. After reordering, the columns $\begin{bmatrix} U_{11}^T & U_{21}^T \end{bmatrix}^T$ contain

the stable invariant subspace of matrix H. Now it is possible to acquire the solution of the AMRE by solving the linear system

$$U_{11}L = -U_{21} \tag{63}$$

The complete algorithm is shown in the Figure 8.

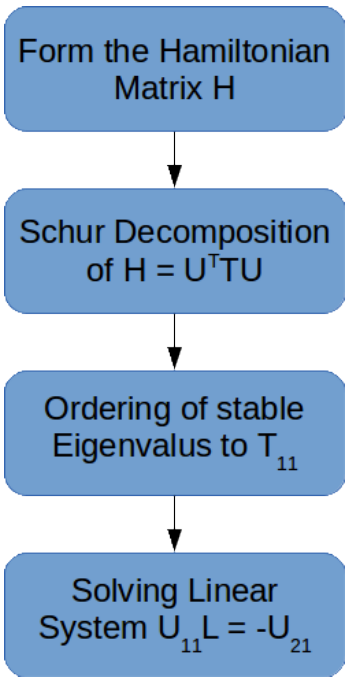

Figure 8: Schur decomposition algorithm

For solving of the linear systems like in (63) there exist several standard methods. For instance, factorization of the U_{11} to an upper triangular via the LU decomposition and then multiple application of back substitution algorithm to the right hand side of the system (Golub and Van Loan, 2012), (Sima, 1996).

Schur decomposition algorithm for finding the solution of the AMRE on average requires around $75n^3$ FLOPs (Laub, 1979).

5.2 Kleinman algorithm

Iterative algorithms require a starting guess which gets refined and nearer to the solution of the AMRE after every iteration. The final result is obtained only after its convergence, that is when the change between two subsequent results is really small. One iteration normally is not computationally expensive. But with the starting guess far away from the solution many iterations might be needed, and consequently requirement of computation operations can be high.

In this category the Kleinman algorithm have stood out with its quadratic and monotonic convergence. It also can guarantee the convergence if the starting guess is stabilizing (Katsev, 2006).

Kleinman algorithm uses the concept of cost matrix, which is shown to be the solution of the AMRE (Kleinman, 1968) after the convergence. This cost matrix monotonically decreases every iteration.

Iteration output is acquired by solving the Lyapunov equation

$$L_i A_k + A_k L_i^T = D \tag{64}$$

where $A_k = A + BK_i$, $D = Q + K_i R K_i^T$ and $K_i = -L_{i-1} B R^{-1}$. This algorithm is also depicted in Figure 9.

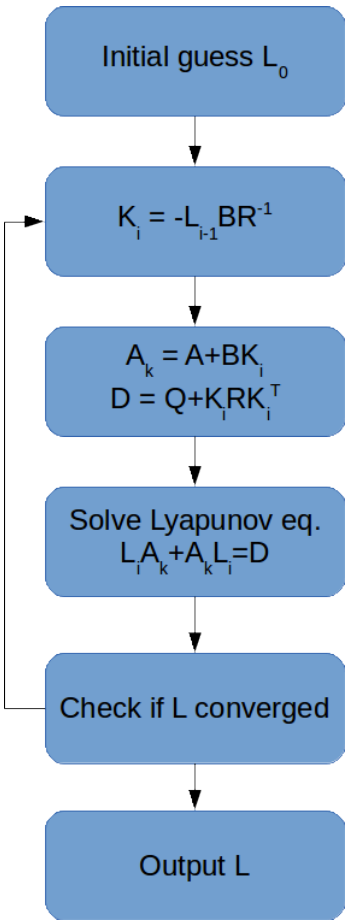

Figure 9: Kleinman algorithm

Lyapunov equation is a special case of a Sylvester equation

$$AX + XB = C \tag{65}$$

when $A = B^T$ (Sima, 1996). Both of them can be solved by rewriting them in a form of regular system of linear algebraic equations, though with large number of unknowns computation requirement is high. On the other hand, using methods with transformation of the matrices A and B to the Schur form like Bartels and Stewart algorithm can be quite efficient.

Kleinman algorithm per iteration requires $6n^3$ FLOPs (Katsev, 2006). This means that as long as the number of iteration is approximately less than ten, performance of the Kleinman method is more efficient than the Schur decomposition algorithm.

5.3 Condition number

Matrix linear algebra problems can be susceptible to small variations in the matrix values (Sima, 1996). This variants are introduced due to the rounding errors caused with the limited precision of the digital processors, and as such can not be avoided. Depending on the matrix values in a problem those variations can affect the solution slightly or greatly. When the small variations result in a small variations to the problems solution the problem is well-conditioned. Contrary, when the small perturbations cause big changes in the solution, then the problems is ill-conditioned. Apparently solutions of the ill-conditioned problems can be quite different than the real solution and thus are undesirable.

Condition number is a measure of the dependence of the solution variations with the problem matrix variations. Calculating it can give the information if the solution is reliable.

6 Simulations

It is obvious that the real environment of the satellite's orbit is not available while the satellites are being designed and constructed. For some modules and satellite parts it is possible to fabricate the environment for testing purposes. This can be really valuable, however it is rarely possible to produce sufficiently precise surrounding and it can be quite costly. Cheap and often suitable approach for testing satellite's functionality is with the computer simulations. Even though sometimes a fabricated environment is available, the computer simulation is still used as an intermediate step toward the validation of the satellite's modules as they might speed up the process. The attitude control methods in this project were developed and tested with one such simulator.

Simulator environment used, Aalborg Toolbox (Jensen and Vinther, 2010), was developed by the students from the Aalborg University for their CubeSat satellite AAUSAT3 project. Parameters of the Aalto-2 satellite have been included in the simulator and control algorithms implemented. Then their performances have been recorded.

6.1 Simulation framework

The simulation framework is developed in Matlab as a Simulink library. Its core consists of Simulink blocks for satellite dynamics, orbit propagation, environment disturbances and models of ephemeris, eclipse, albedo and magnetic field. Additionally simulator provides models of the sensors and actuators and mathematical functions for working with vectors, matrices and quaternions. These simulator modules will be briefly explained.

Simulation time block Time generating block is needed to set pace of simulation and to synchronize all other blocks. Time is used by other blocks to calculate the satellite's position in orbit and the relative positions of the Earth, Sun and Moon. Initial time and time decimation can be configured.

Spacecraft dynamics block This is a Simulink block responsible for tracking of the satellite's attitude. It implements the dynamic satellite differential equations (14). As the input it takes the initial angular velocity, initial body attitude, satellite's inertia matrix, controller frame, disturbance torques and the control torques. By the integration of differential equations, supplied with the mentioned inputs, this block outputs satellite's current attitude and angular velocity.

Orbit propagator block Determining the satellite's position in orbit is done with the SGP4 orbit propagator model. SGP stands for Simplified General Perturbations. The SGP4 is commonly used for the satellites in low Earth orbits. The model adds up the effects of gravity of Earth, Moon and Sun, Earth's oblateness and atmospheric drag (Hoots et al., 1980). Information given by this block is needed for determining the tracking reference point and the vector of Earth's magnetic field at the position

of satellite. Its inputs are the Two Line Elements (TLE) and the current time. Outputs of the block are satellite's position and velocity.

Ephemeris model block Rotation of the Earth is needed for determining the Earth's magnetic field model. Also, relative positions of Sun and Moon are used in calculation of disturbances affecting the satellite. Input to the block is current time and outputs are Earth's rotation and positions of the Sun and the Moon.

Eclipse model block This block gives as its output the information if the satellite is in the eclipse. For inputs it takes the positions of the Sun and satellite in $\mathcal{F}_i$.

Earth albedo model blocks Taking in as inputs the rotation of the Earth, Sun's position and position of the satellite, this block outputs the reflectivity matrix. Albedo is mainly used for the sun sensor simulation.

Magnetic model block Earth's magnetic field is generated via the International Geomagnetic Reference Field (IGRF) numerical model (Finlay et al., 2010). The model is generated with data collected at the magnetic observatories around the world, various surveys of the magnetic field and the data obtained from satellites. This model can output the Earth's magnetic field vector values since 1900 until present. Model is updated every five years, and the latest model is of 11th generation, IGRF11. It is valid until the 2015. The Simulink block takes the satellites position and the rotation of the Earth as inputs and the output is the vector of the magnetic field at the satellite's position.

Environment disturbances block This block encompasses the effects of disturbances described in section three, including atmospheric drag, solar pressure, gravity effects and magnetic residual. Outputs from this block, disturbance torques and disturbance forces, affect the attitude and orbit parameters of the satellite. Inputs are positions of the Moon, Sun and satellite, satellites attitude, velocity, mass, inertia matrix, center of mass, center of pressure and controller's frame of reference. The magnetic field vector and eclipse are also taken as inputs.

6.2 Simulation parameters

Here are listed the satellite parameters used in the computer simulations.

Orbit parameters:

- Altitude - $380km$ and $200km$
- Inclination - $98.6°$
- Eccentricity - 0.064
- Time - December 2015

Satellite's parameters:

- Weight - $2000g$
- Size
  - X - $207mm$
  - Y - $100mm$
  - Z - $100mm$
- Moment of principal inertia matrix

$$I = \begin{bmatrix} 0.007 & 0 & 0 \\ 0 & 0.014 & 0 \\ 0 & 0 & 0.015 \end{bmatrix} kg \cdot m^2 \tag{66}$$

- Center of mass - 0, 1 and 2 cm from geometrical center toward the ram side
- Maximum magnetic dipole moments of magnetorquers - $\pm 0.2 \frac{Nm}{T}$

Environment parameters:

- Atmospheric density - 9.955e-12$\frac{kg}{m^3}$ ($380km$) and 3.162e-14$\frac{kg}{m^3}$ ($200km$)

Different parameter sets have been used to simulate and test the control methods. Among them, two orbit altitudes are selected, 380 km for the orbit at the beginning of the mission and 200 km for the orbit at the end of the mission. For both of the altitudes, appropriate atmospheric density parameter is provided. Center of mass is varied over three locations during the passive atmospheric drag stabilization tests, to examine its influence on the satellite's attitude stability. Though for the rest of the tests, the value of 1 cm is selected. Although the orbit eccentricity for the satellite mission is maximally 0.04, a value of 0.064 is used in the simulation as this is the smallest eccentricity available with the used orbit propagator. Other parameters related to the control methods will be explained in the subsections of the respective method.

The tests are simulated for the duration of 170 minutes for the detumbling, and 25 hours for the rest of the tests. These times are approximately 2 and 15 revolutions of the satellite around the Earth for the mission orbits.

During the tests, the control methods have been provided with the real values of the satellite's attitude and angular velocity and orbit's angular velocity. However, these values will be provided by the Attitude Determination System during the mission, and they will incorporate certain amount of noise. That is, the accuracy of the values provided to the control methods will be limited. This implies that the accuracy of the ACS attained in the tests is better than the accuracy possible

for the mission. On the other hand, magnetic field vector values are emulated to resemble the values from the magnetometer sensors. Noise and bias is added to the real values of the magnetic field. Nonetheless, current setup can be used for rough evaluation of control methods and their comparison. Unavoidably, tests including the ADS need to be performed as soon as the ACS and ADS integration is possible.

6.3 Passive stabilization

Six tests have been performed for the passive stabilization. First three tests were done at the altitude of 380 km, with the three different displacements of the center of mass from the geometrical center. Other three tests are for the altitude of 200 km, again with the same center of mass displacements.

Plots of the attitude angle errors are shown for each test, and the table 1 shows the magnitudes of the atmospheric drag disturbance torques experienced by the satellite. Also, two plots of the angle error histograms are shown, comparing the tendency of the satellite toward the desired attitude.

	0 cm	1 cm	2 cm
380 km	3×10^{-7}	5×10^{-7}	7×10^{-7}
200 km	1×10^{-5}	1.5×10^{-5}	2×10^{-5}

Table 1: Atmospheric drag disturbance torque magnitudes, Nm

In the table, the magnitude increases practically linearly with the increase of the center of mass displacement. And there can be seen a great increase at the lower altitude, with the factor of approximately 30.

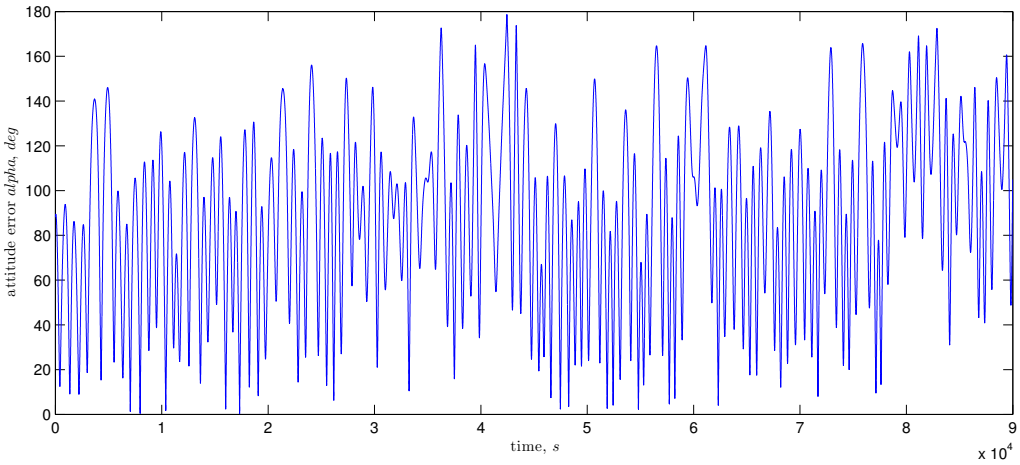

Figure 10: Angle error, displacement 0 cm, altitude 380 km

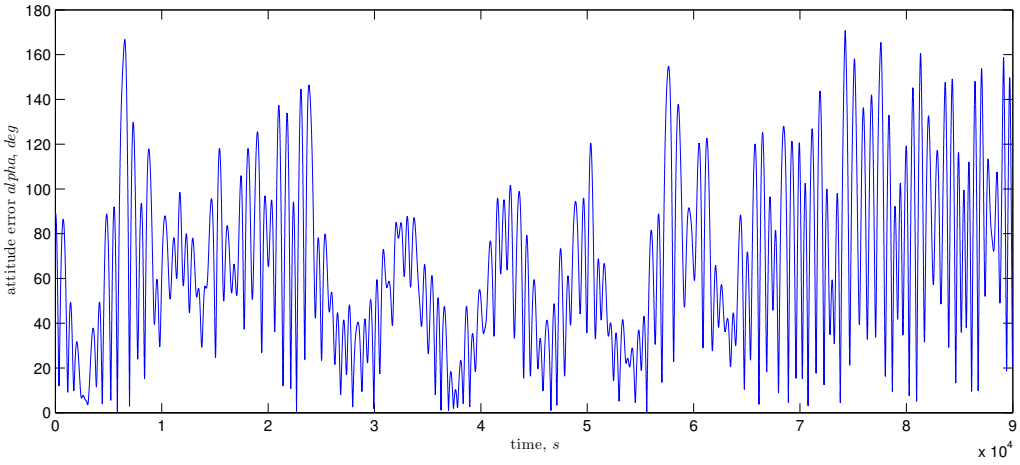

Figure 11: Angle error, displacement 1 cm, altitude 380 km

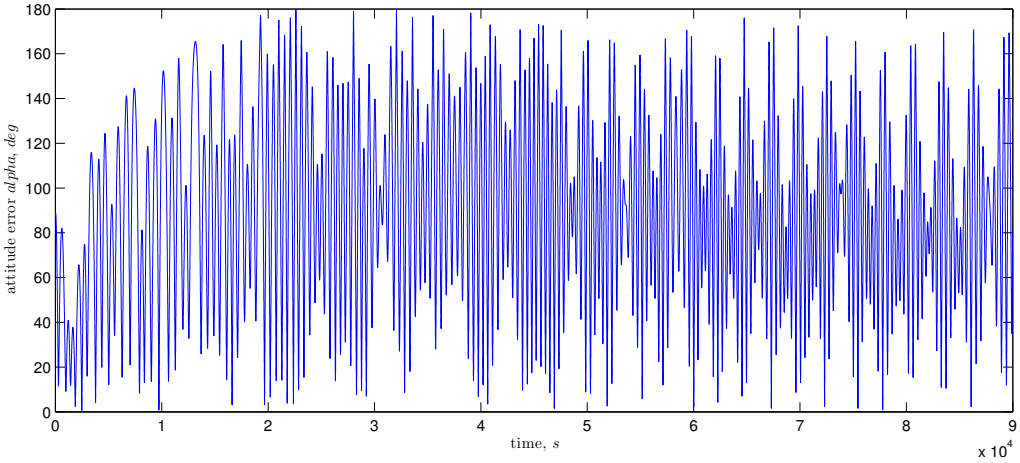

Figure 12: Angle error, displacement 2 cm, altitude 380 km

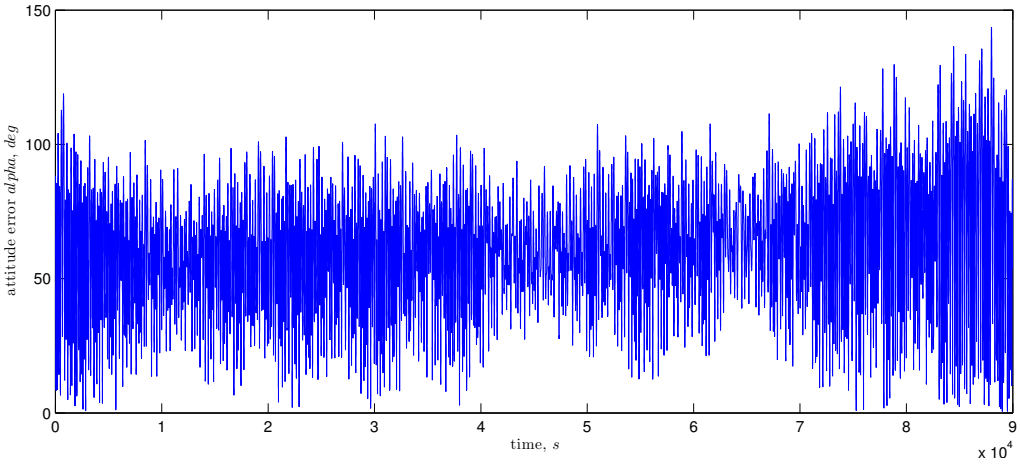

Figure 13: Angle error, displacement 0 cm, altitude 200 km

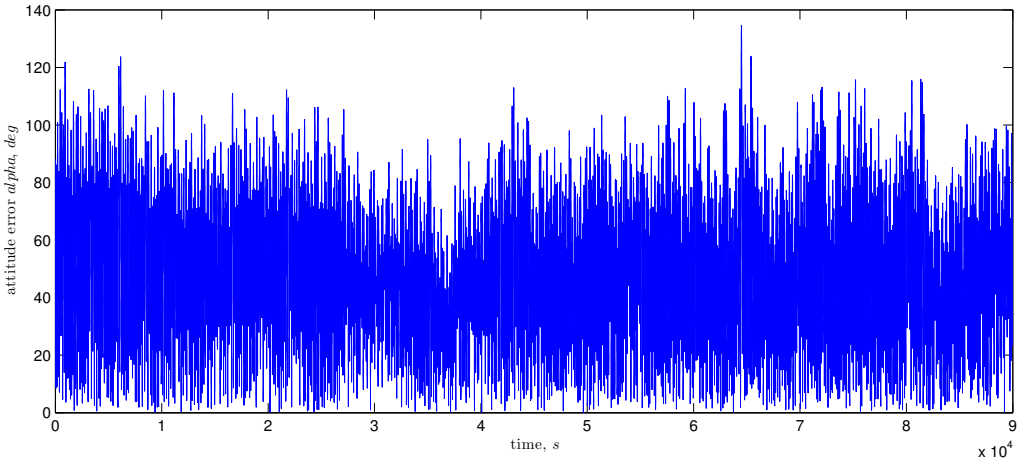

Figure 14: Angle error, displacement 1 cm, altitude 200 km

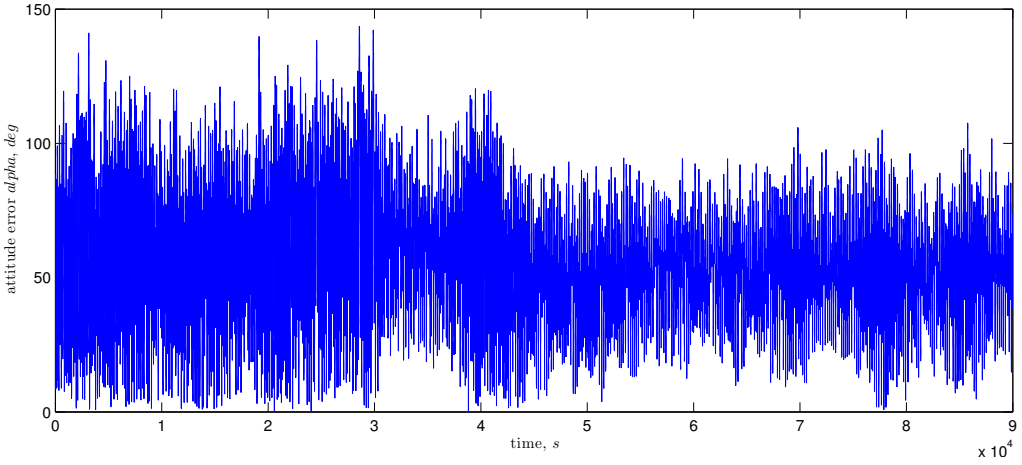

Figure 15: Angle error, displacement 2 cm, altitude 200 km

Influence of the displacement of center of mass on the stability can be seen on the figures 10, 11 and 12 for 380 km, and on the figures 13, 14 and 15 for 200 km. Displacement of 1 cm shows the greatest shift toward the lower attitude angle errors for the higher orbit. The stronger torque with the greatest displacemnt induces larger oscillations, and thus the stability worsens. The lower orbit have negligible effect with the position of the center of mass as the torques are strong and they provoke rapid oscillations.

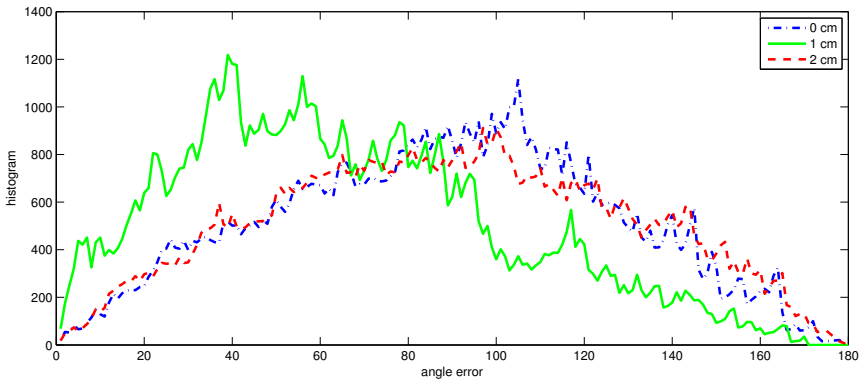

Figure 16: Angle error histogram, altitude 380 km

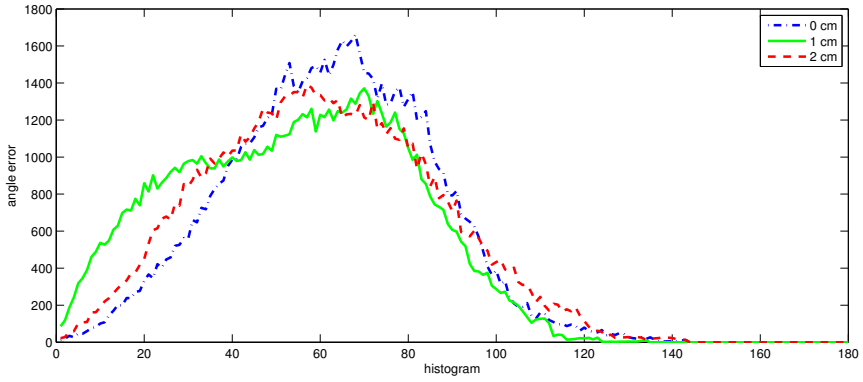

Figure 17: Angle error histogram, altitude 200 km

Histograms of the attitude angle errors in figures 16 and 17 show more closely the tendency of the satellite's attitude toward the desired position depending on the atmospheric disturbance torque.

6.4 Detumbling controller

Detumbling control is tested only at the altitude of 380 km, as this is nearly the deployment orbit. Three control gains are tried, 5000, 10000 and 15000. Initial angular velocity is set to $15°/s$ around all three body axes. Plots show the angular velocity around principal axes of the body..

Satellite's angular velocity is slowed down in approximately 150, 80 and 60 minutes for the control gains 5000, 10000 and 15000, respectively. This can be seen in the figures 18, 19 and 20.

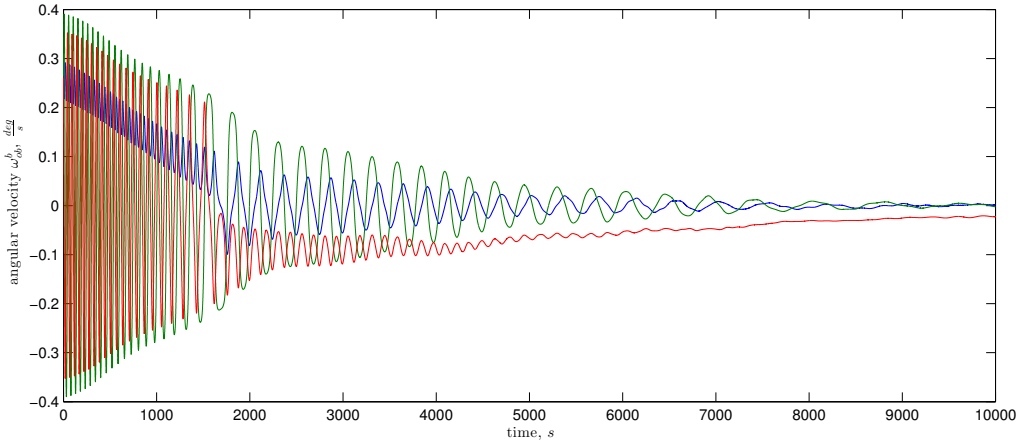

Figure 18: Angular velocity, K = 5000

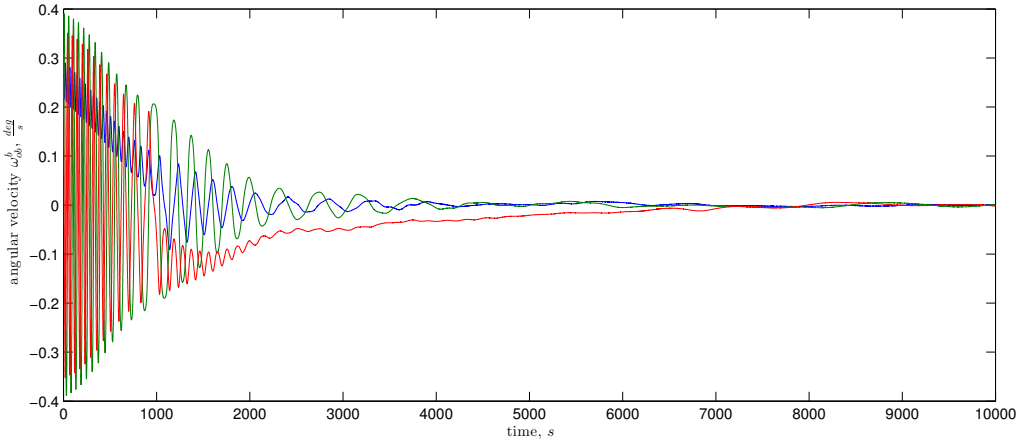

Figure 19: Angular velocity, K = 10000

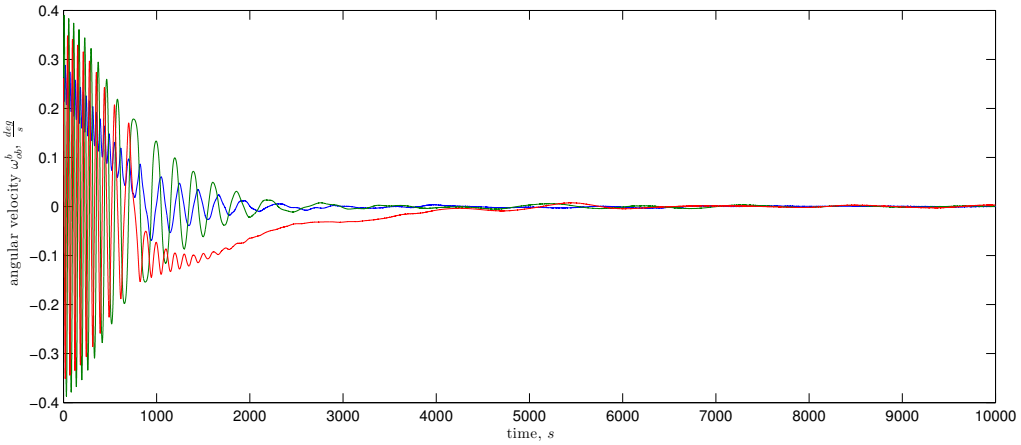

Figure 20: Angular velocity, K = 15000

6.5 PD controller

Angle errors of the PD control tests for both of the orbits are shown in figures 21 and 22. The values of the P and D gains were determined empirically. Best combination proved to be P $= 2.5 \times 10^{-10}$ and D $= 1 \times 10^{-4}$. Higher damping is needed to subdue the oscillations invoked through the atmospheric drag torque. Too strong of the proportional gain invokes tumbling of the satellite and its effect on stability is minimal. Thus the derivative gain is substantially larger than the proportional gain.

This control method did not met the requirement of the mission. For both orbits, the angle error goes beyond 15°. The higher orbit angle errors are greater and they can reach 40°, though there are occasional periods with stability near the requirement error. For the lower orbit angle errors are smaller, and they can reach 20°. Oscillations around the desired attitude are higher.

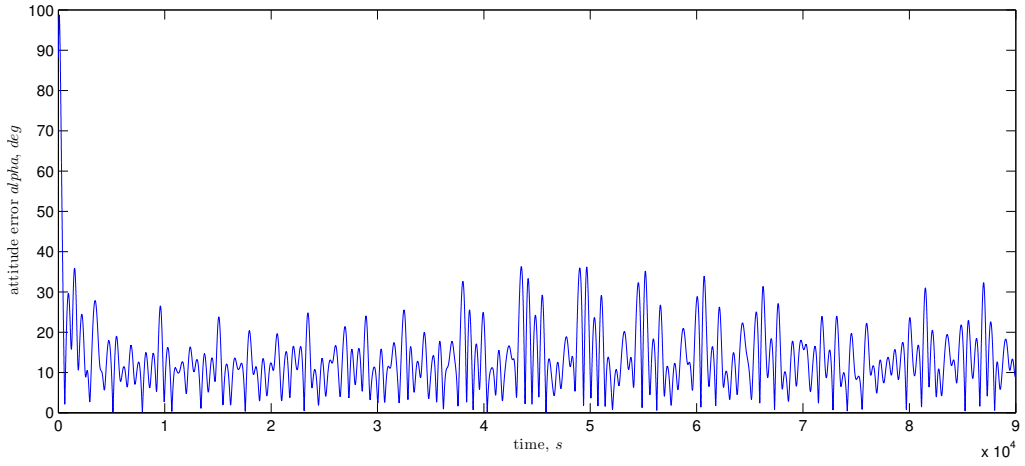

Figure 21: PD control, angle error, 380 km

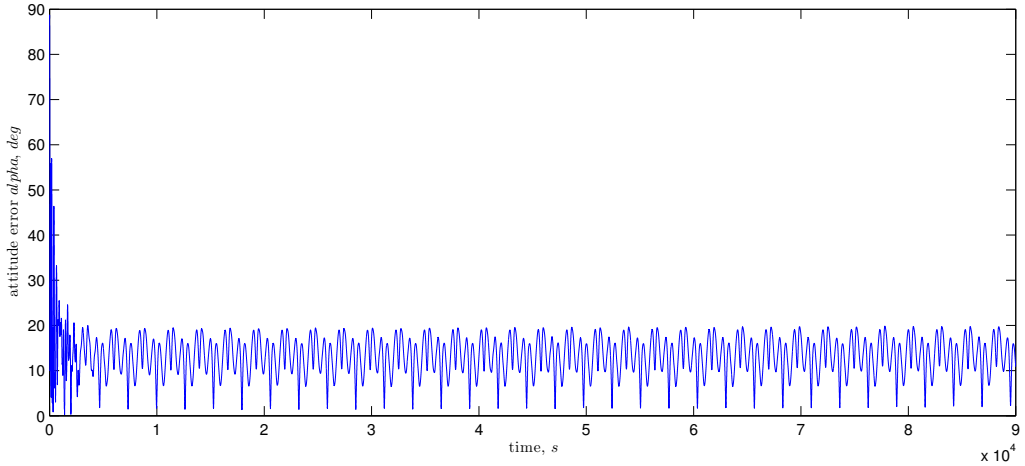

Figure 22: PD control, angle error, 200 km

Histograms show that the orbit with the altitude of the 380 km was below 15° for 66.52% of the time, and at the altitude of 200 km it spent 59.33% of time within permitted angle error region.

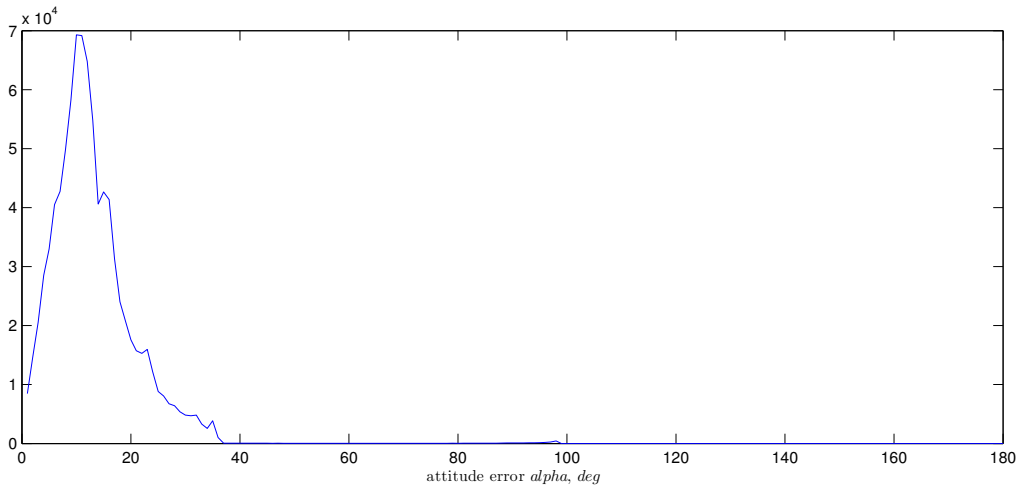

Figure 23: PD control, angle error histogram, 380 km

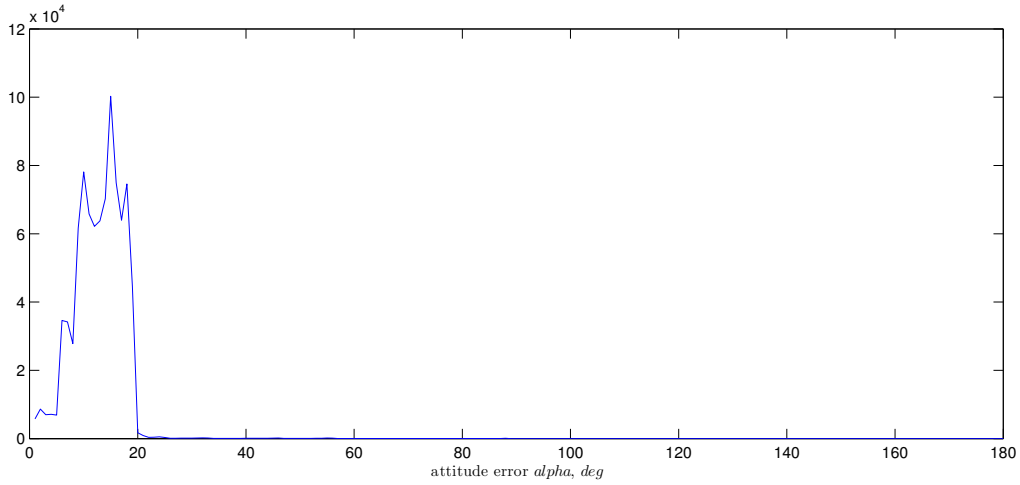

Figure 24: PD control, angle error histogram, 200 km

6.6 LQR controller

Same tests for the PD control were also done for the LQR controller. The Q and R matrix values were empirically selected too: $Q = diag([10\ 100\ 100\ 10^{-5}\ 10^{-5}\ 10^{-5}])$, $R = diag([100\ 100\ 100])$. As the rotation around the satellite's x axis is unimportant for attaining the desired attitude, cost for its error is smaller than for the other two axes.

Figure 25 shows the performance of LQR at the altitude of 380 km, where the attitude error was maintained below 15° most of the time, though there are regular

short oversteps beyond 15°, which can go as high as 40°. On the histogram of figure 27 it can also be seen that the majority of time the attitude is below 15°, precisely 80.56% of the time. At the beginning of the angle error plot there is a period with high angle errors, because the LQR is linearised near the desired attitude and is less efficient far away from it. The orbit of 200 km shows better performance, the LQR control keeps the attitude within the requirement for a larger time percentage, as can be seen on the figures 26 and 28. Time spent under 15° of attitude error is 88.34%.

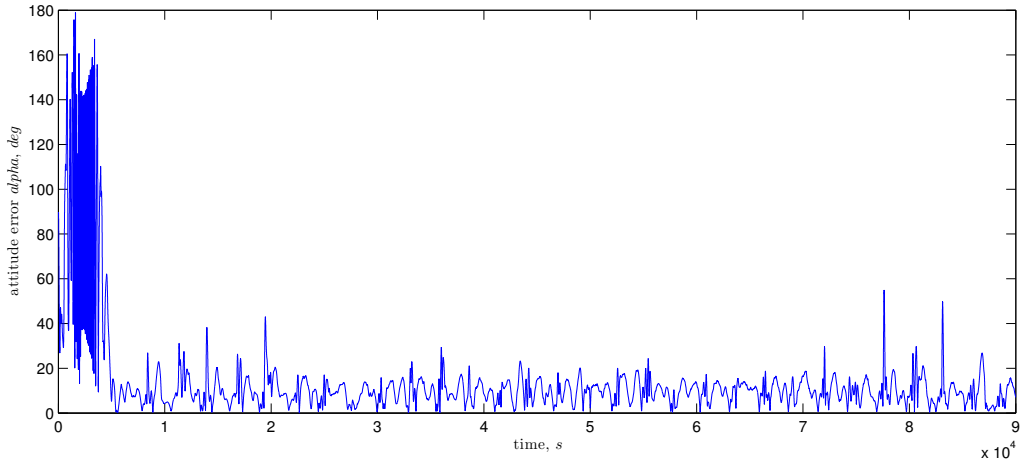

Figure 25: LQR control, angle error, 380 km

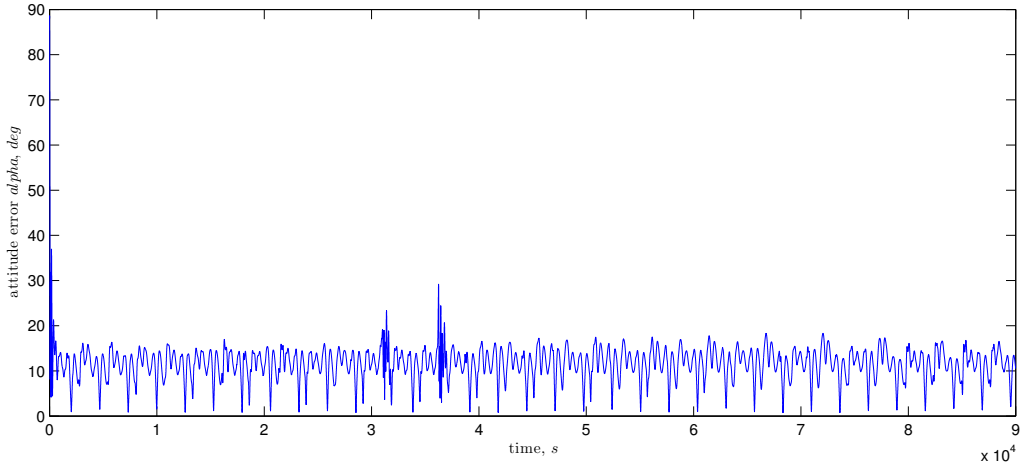

Figure 26: LQR control, angle error, 200 km

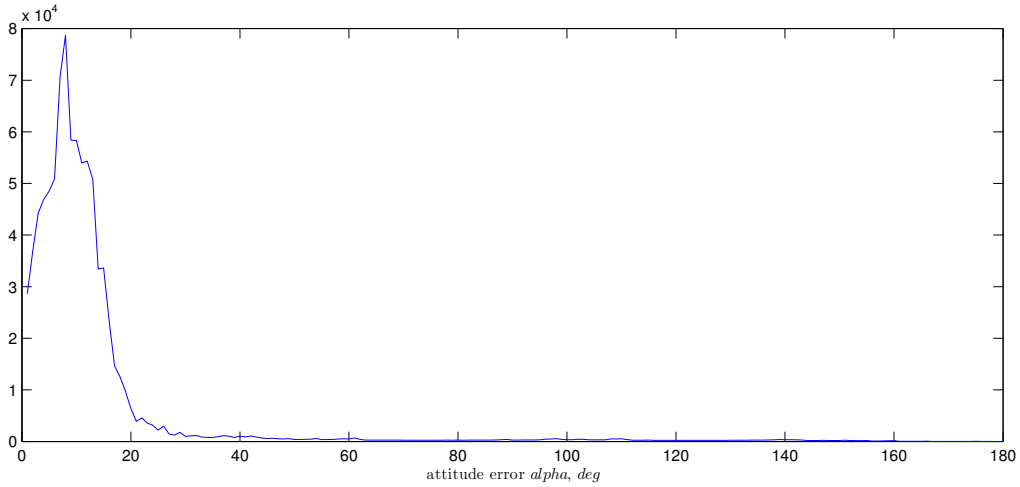

Figure 27: LQR control, angle error histogram, 380 km

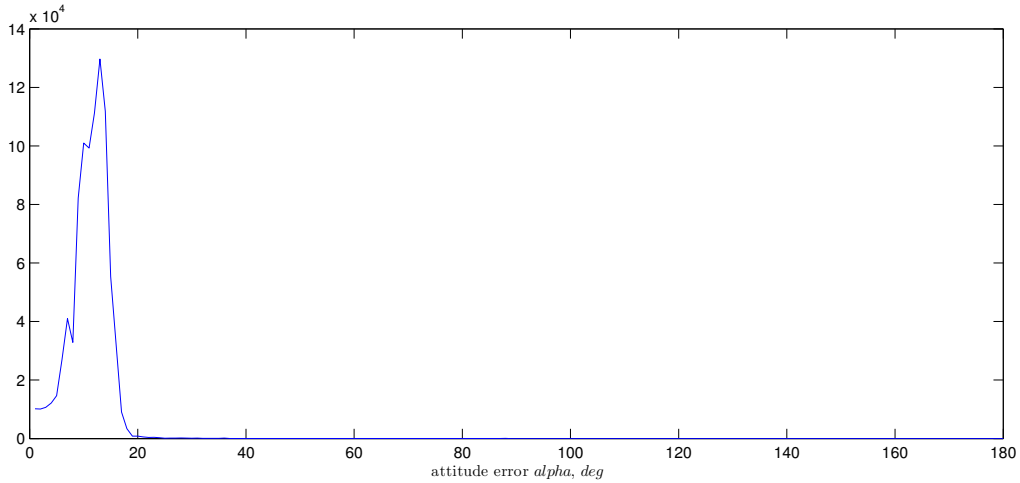

Figure 28: LQR control, angle error histogram, 200 km

6.7 SDRE controller

Two altitudes, 380 and 200 km, were used for testing of the SDRE as well. Again, the best pair of Q and R matrices were found empirically, which are the same as for the LQR. Matrix values are $Q = diag([10\ 100\ 100\ 10^{-5}\ 10^{-5}\ 10^{-5}])$ and $R = diag([100\ 100\ 100])$

SDRE control performed slightly worse than LQR, as the satellite's attitude was outside the desired region for longer time. Attitude was inside the desired angle error region for 72.58% of the simulation time. There were also occurrences of greater angle errors, as big as 70°. This happens as the SDC system matrix is calculated online from the system state, and the solution of the AMRE is more susceptible to variations in this case. At the lower orbit, as seen in the figure 30, SDRE control

performed much better, keeping the attitude within wanted region for 82.47% of the simulation time.

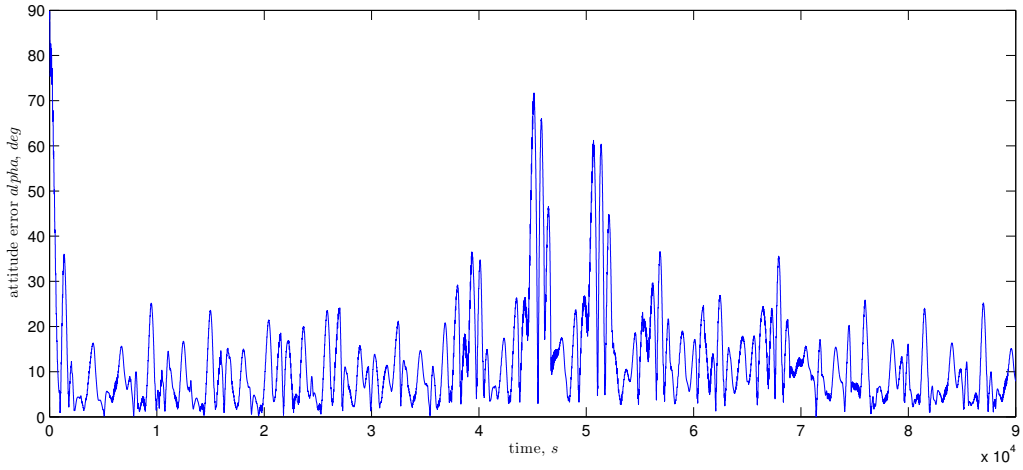

Figure 29: SDRE control, angle error, 380 km

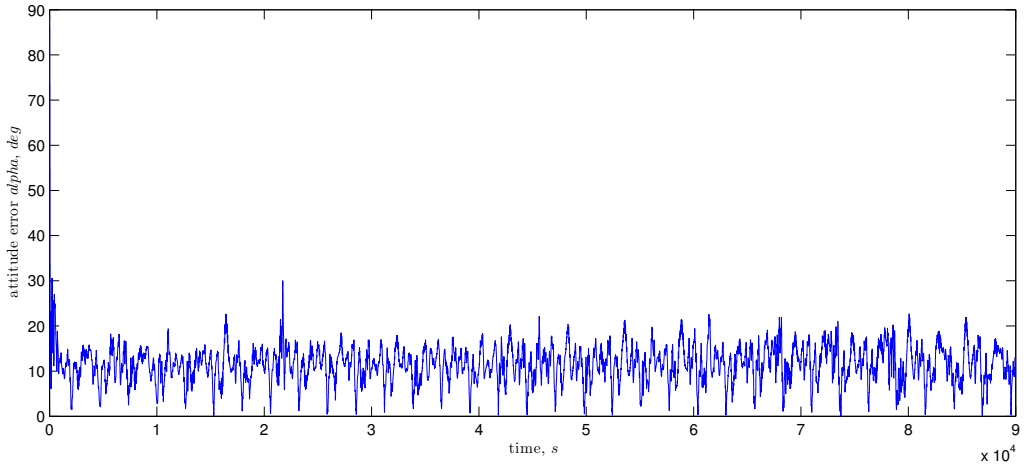

Figure 30: SDRE control, angle error, 200 km

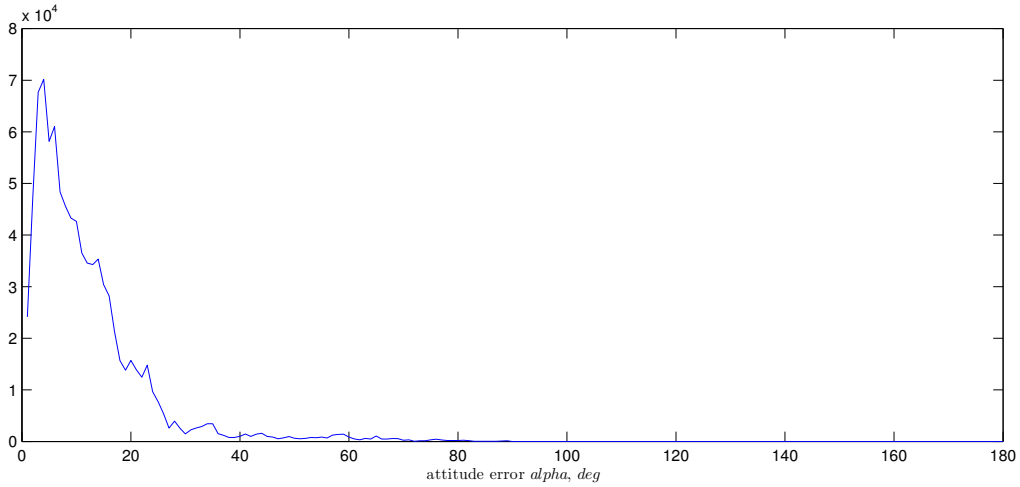

Figure 31: SDRE control, angle error histogram, 380 km

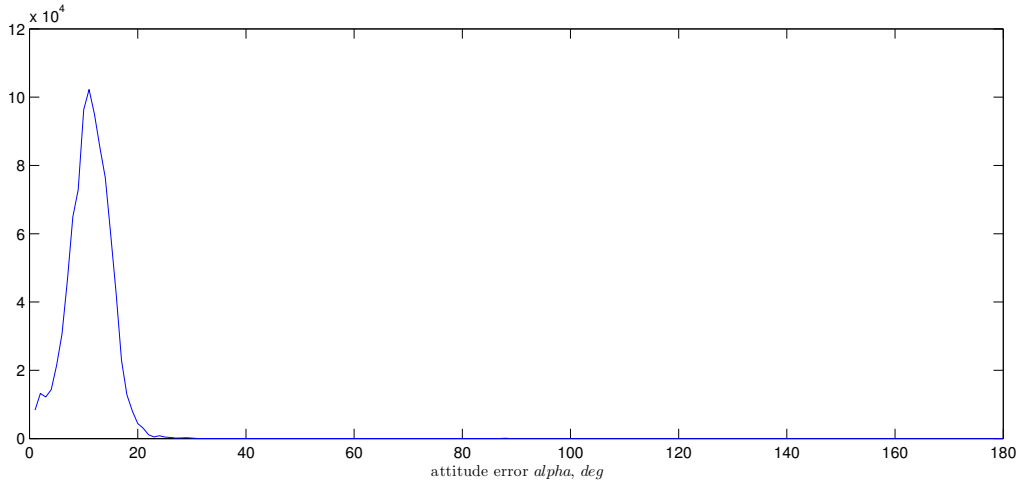

Figure 32: SDRE control, angle error histogram, 200 km

6.8 Control signal update frequency

In all of the tests above, update frequency of the control signal was set to 1 Hz. Additionally, this setting was gradually changed to evaluate how the change affects the control and to discover the requirements for proper control.

Effects of the update frequency change were barely noticeable with any increase and also until the decrease to 0.2 Hz. While decreasing even more under 0.2 Hz, until 0.1 Hz, some performance degradation appeared. However, even with 0.1 Hz update frequency, the satellite could be controlled closely to the performance of higher update frequency. Finally, 1 Hz have been selected as the desired frequency for control signal update, leaving some margin before the noticeable change in controllability.

7 Implementation

Final goal of this project is to have the selected attitude control method implemented for the satellite's attitude control system which consists of the onboard computer, electric drivers for magnetorquers and magnetorquers. The control software is supplied with data acquired from attitude determination system and magnetometers. This section will describe the hardware components, software infrastructure and libraries, and developed algorithms used for controllers.

As have been seen in the simulation results, the LQR controller is able to keep the state closer around the reference point, but the SDRE is performing better when the attitude is further from it. As the difference between these two algorithms is just in the system matrix A, it is possible to include both of them and select the appropriate one depending on the distance from the reference point. Switching will also effect in the need to recalculate the solution of Riccati equation with the Schur method.

7.1 Onboard computer

Core of the onboard computer is the Texas Instruments Safety-Critical Applications RISC Flash Microcontroller RM48L952PGE. Central processing unit of this microcontroller is 32-Bit ARM-Cortex R4F with single and double precision floating point unit. Supported system clock is up to 220MHz and the CPU can attain 1.66 DMIPS/MHz. The microcontroller have integrated 3MB of program flash memory, 256kB of RAM memory and 64kB EEPROM of flash memory. It is also equipped with timer modules, modules for input/output like GPIO, various serial communication modules and other.

Communication with the electric driver for magnetorquer is interfaced through the I2C communication protocol.

7.2 Software infrastructure

Onboard computer will be assigned with several tasks, constituting a functional modules, amongst which is the attitude control. In embedded systems one can find several ways of maintaining tasks for multipurpose systems, often being the SuperLoop, Event based structures and Time based scheduling. Similar classification can be found in Rao et al. (2009).

The simplest is SuperLoop, it iterates over several functions representing each task of the system. Though this approach is quite straightforward and with minor overhead, it comes with some drawbacks. Managing priorities of the tasks is not possible and it is hard to equally distribute the processing time. Also the task functions themselves have to take care of letting other tasks to continue their work, which impose additional complexity on them.

Event based structures are usually implemented with finite state machines, which complexity depends on the number of tasks and possible events, or simpler polling loops. Tasks are invoked only after a specific event happened, it can be arrival

of message over a communication line or it might be a passing of defined amount of time. This kind of software structures introduce flexibility to task management, however it is mainly suitable to reactive systems with tasks of short execution times.

Time based scheduling can offer a lot of flexibility, task control options with priorities and real time capabilities. Tasks of the same priority are allocated with the same processing time slots. Task management abstraction allow tasks to be devoted to the specific problems, without worrying about other tasks. This approach have a noticeable overhead, however the task execution is arranged in a way to maximally exploit the processing time. FreeRTOS is a real time operating system that implements a time based scheduler, and it was chosen for managing the tasks of the Aalto-2 satellite.

The attitude control algorithms extensively use matrices in their calculations. A mean for easy manipulation of matrices and need for matrix operations is obvious. Various algorithms from linear matrix algebra are also needed. Meschach library offers many of the needed operations and algorithms written in C programming language, which is convenient for this project.

7.2.1 FreeRTOS

FreeRTOS is an operating system with real time capabilities and is designed to be small enough to run on microcontrollers (Real Time Engineers Ltd., 2014). More than thirty processor architectures are supported, and its portable source code makes it possible to easily extend the support to additional architectures. It offers preemptive, cooperative and hybrid scheduling options with support for task priorities. Several inter task communication, synchronization and memory allocation mechanisms are included. Version of the FreeRTOS used is 8.0.1.

Purpose of the scheduler is to manage task execution and distribute processing time according to the task priorities. On a single core processors only one task can be active at a time, and to allow for parallel execution, scheduler frequently cycle through the list of tasks, giving them equal time slots. Scheduling scheme that FreeRTOS uses by allocating the same time amounts for each task is called Round-Robin. The tasks with lower priorities are executed only when all the tasks with higher priorities are unable to run. There are several states a task can be in. Currently active task is in a running state. A task is in the waiting state is able to run but is in a queue, waiting for its time slot. Blocked tasks are waiting for some resource or a signal to continue their work and are exempt from scheduling until their requirement is fulfilled. Tasks can also be manually suspended, and remain in that state until they are manually resumed.

A task in FreeRTOS is a dedicated portion of code with its own context. The context preserves the values of all registers within processor used for the task execution and the memory addresses of the task's stack space. Scheduler is able to exchange the context of currently active task with the context of a tasks in a waiting list. Tasks themselves do not perceive this change.

For the synchronization FreeRTOS provides multiple options. Binary semaphores are resources that can be reserved. If the binary semaphore is free, reservation is

possible right away so the task can continue with its job, and if it is already reserved the task have to wait until it is freed. Mutexes are similar to binary semaphores, they additionally have the priority inheritance mechanism, which comes in handy when there is a possibility for a deadlock. Counting semaphores can be reserved a defined number of times, allowing several tasks to proceed at the same time.

Queues are used for safe and easy information sharing between tasks. Data packets can be inserted into a queue. The packets can be taken from the queue in the same order they have been inserted. Queues take precaution that the data is transfered correctly.

Timers allow for some tasks to be scheduled at a given time in the future. They can be set to repeat with a defined period, or just executed once.

Event groups implement the event system as another synchronization option. Each group can define several events and tasks can await for the events to happen. Task which await for the events is blocked until at least one of the events from the group is signaled, then the task is unblocked by the scheduler and supplied with information of the signaled events.

In real time systems memory handling require some precautions. In general, dynamic memory handling should be minimized. To preserve the real time capability the FreeRTOS had to include only simple algorithms for memory management. Three options are provided. The simplest memory management algorithm provided is able only to allocate a memory space and it can never deallocate reserved memory portions. This option gives the quickest and most predictable response, however it can only be used in systems with well defined and constant memory requirements. Second option includes the possibility for freeing of allocated memory spaces, though this freed portions is not recombined with free memory and can be reused only if new allocation requires the same or smaller memory sizes. Second option is adequate for applications that frequently allocate and free constant memory blocks. Third option can combine freed memory blocks into free memory, still the allocated memory blocks can not be moved so there remains some memory fragmentation, and with inappropriate handling a lot of memory can be wasted. The third option will be used in the project.

7.2.2 Meshach library

Meschach is numerical library for matrix and vector calculations and linear algebra algorithms (mes, 2014). It is developed at the School of Mathematical Sciences of Australian National University. Library is written in C programming language and consists of data structures and functions for representing and handling of matrices and vectors. Number of standard matrix and vector operations are included and it gives the basis of linear algebra algorithms. Meschach provides full control of memory usage to user which is valuable for embedded systems. Used version of the library is 1.2b.

Main data structures are vectors and matrices. Various operations can be performed on them like addition, subtraction, vector-vector, matrix-vector and matrix-matrix multiplications, transposition, inversion, Giddens rotations and many more.

As for the linear algebra algorithms, Meschach provides several different matrix factorizations like LU, QR and Schur decompositions. Different solvers are also available for solving of systems of linear equations, for instance *LUsolve* and *QRsolve*.

7.2.3 Tasks structure

Satellite's software system consist of several tasks, and for attitude control module tasks of importance are system's Main task, the ControlSignal task, TorqueDriver task, SensorReading task and StateDetermination task. Figure 33 shows the tasks related with the attitude control system and their data passing connections.

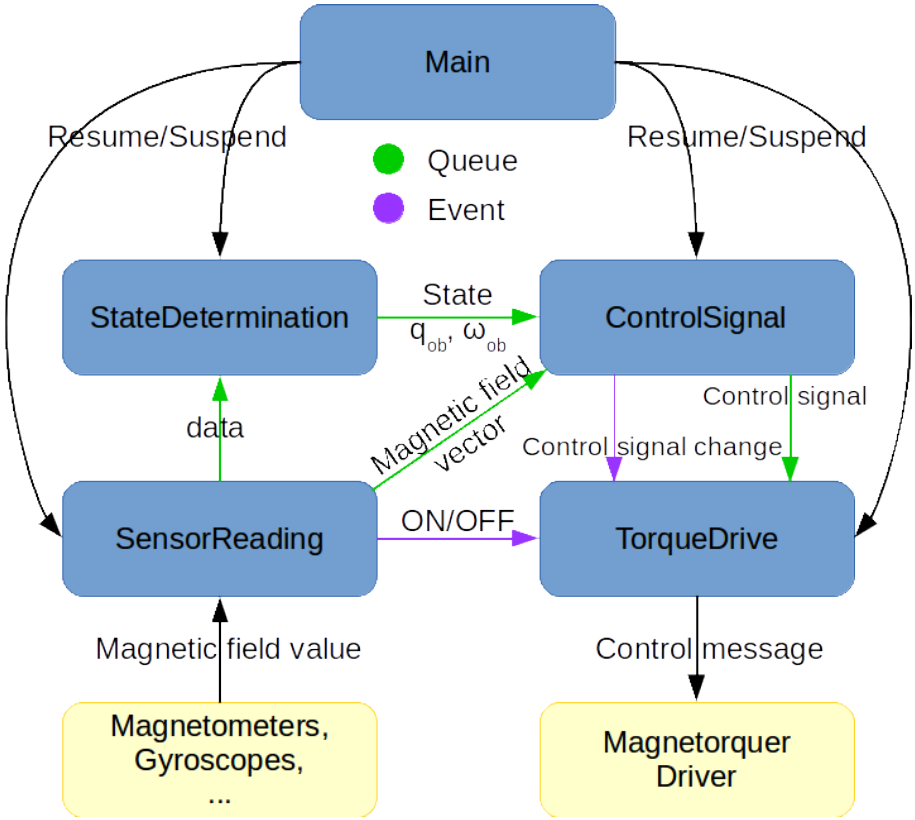

Figure 33: Tasks in the system and their connections

Tasks exchange the data through one element queues. The tasks are always in need only of the most recent values supplied to them. For this reason the queues are limited to only one element. If new data arrives to the queue before the old one was read, it will simply be overwritten when the FreeRTOS function *xQueueOverwrite* is used for sending. This way a task will never have to filter messages for the newest data. And if there were no new data since the last check, the queue will be empty so the task can be aware when there is no need to recalculate any new values.

System's Main task control the satellite's working mode and manages other tasks. After the satellite is deployed in the orbit, startup phase is initiated and Main task will take care of bringing up of all the necessary systems. Attitude control related tasks will be started by the Main task when all prerequisite phases are complete.

Determination and control of the attitude are crucial for satellites proper functioning during its mission phase, and it is assumed that these tasks will be active throughout the rest of the mission. Nevertheless, activity of the tasks remain under control of Main task in case of need for maintenance. Tasks are enabled with the FreeRTOS function *vTaskResume* and disabled with *vTaskSuspend*.

TorqueDriver task sends the control messages to the magnetorquer driver hardware. The task awaits for the events on an event group. The event group have three events defined, "Control signal change", "ON" and "OFF". If the "Control signal change" event is raised, task will receive a new values of the control vector through queue that was sent by the ControlSignal task. The new control vector values will be forwarded to the hardware driver for magnetorquers. "ON" and "OFF" events will send messages to the hardware driver to turn on and turn off the magnetorquer actuation. Magnetorquers are supposed to be actuated as often as possible, however magnetic moment generated by them distorts the magnetic field around the satellite. Thus, magnetometers are not able to retrieve correct magnetic field measurements while the magnetorquers are actuated. SensorReading task will raise the "OFF" and "ON" events when it needs to read the magnetic field measurements. Diagram of the task is shown in Figure 34.

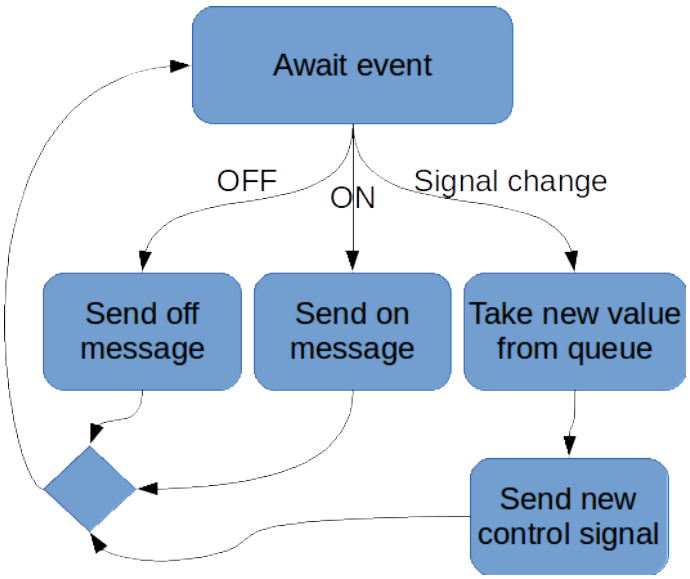

Figure 34: TorqueDriver task diagram

Purpose of the ControlSignal task is to calculate the control vector with which to drive magnetorquers. Task is controlled with the timer which executes it regularly every calculation period. It receives the state values, attitude unit quaternion and angular velocity vector, from the StateDetermination task and magnetic field vector from SensorReading task over one element queues. At the task's execution start it checks the queues for new data. If new data is supplied, the task can calculate new control vector values, send them to the torque driver task through the queue and signal the event to inform it about the change. ControlSignal task then can block itself until the next cycle. Depending on the received data from the state

determination task, control vector will be calculated either by the detumbling, LQR or SDRE controller method. If angular velocity is higher than certain threshold, detumbling controller will be used, LQR controller is used when the state is close to the reference point, otherwise SDRE controller calculates the control vector values. Task diagram is presented in Figure 35.

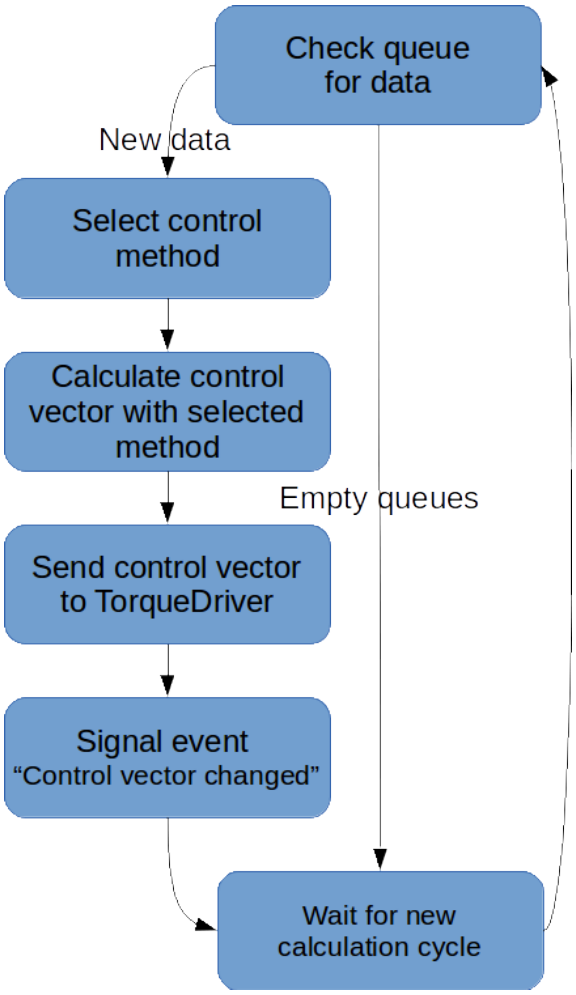

Figure 35: ControlSignal task diagram

SensorReading task collects the data from all the sensors of the system. It is designed so it can have different reading periods for different sensors. The task is executed by the timers that define reading periods. After a value is acquired from a sensor, it is sent to the receiving tasks over queues, and if the task awaits for the event, the event is signaled. In the case of reading of magnetometers, the TorqueDriver will be notified with events to turn off the actuation before the sensor is read and to turn the actuation back on after the reading is finished.

StateDetermination task provides the state data to the ControlSignal task and other tasks that need this data. Data is sent through one element queues. This task receives the data from the SensorReading task also through the one element queue. Task is periodically executed by the timer.

7.3 Implemented numerical algorithms

Kleinman and Schur decomposition methods have different approaches for finding the solution to the AMRE, so they are using different linear algebra algorithms. Some of the algorithms used in those methods already exist in the Meschach library, however some are not included. Thus the algorithms that are missing have been additionally implemented. This numerical algorithms that are used in Riccati solvers are explained.

Main steps in the Schur decomposition are the Hamiltonian matrix formation, Schur factorization, ordering of the eigenvalues and solving of the linear system. To form the Hamiltonian matrix it is needed to copy and multiply matrices, which is simply done in Meschach. Schur factorization can be performed with the function *schur* and for solving of the linear system the functions *QRfactor* and *QRsolve* are employed. Algorithm for ordering of the eigenvalues of the Schur decomposition can not be found in Meschach and therefore it was additionally implemented.

Kleinman algorithm iteration consists of forming of the matrices A_k and D, which is done by simple multiplication, addition and transposition of matrices, and their insertion in a Lyapunov equation. Then the solution of the Lyapunov equation is the solution of AMRE when it converges. Meschach does not contain any function for solving of the Lyapunov equation, so implementation of this algorithm was needed. This algorithm is described in the subsection below.

All of the implemented algorithms are explained in details in the book *Algorithms for Linear-quadratic Optimization* (Sima, 1996).

7.3.1 Ordering of Schur eigenvalues

The ordering of the eigenvalues in a Schur factorization $H = UTU^T$ is done by iterations over the diagonal matrix blocks of T, beginning at the uppermost block. When the two adjacent blocks are encountered where the upper one is with positive real value and the lower one is with negative real value, their places on the diagonal are swapped. Iterations are performed until there are no more possible swaps.

The proposed algorithm (Sima, 1996; Bai and Demmel, 1993) assumes that the swapping blocks have no eigenvalues in common and that the 2x2 blocks are in standardized form. Blocks are in standardized form when its diagonal entries are equal and the product of the off-diagonal entries is negative.

Matrix S is a submatrix of T that consists of the entries of the two adjacent diagonal blocks that need to be swapped (S_{11} with size pxp and S_{22} with size qxq) and the off diagonal part (S_{12}) next to the diagonal blocks.

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \tag{67}$$

Diagonal blocks S_{11} and S_{22} can be swapped by the orthogonal matrix Q of the size $(p+q)$x$(p+q)$if and only if (Bai and Demmel, 1993)

$$Q^T \begin{bmatrix} -X \\ I_q \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{68}$$

where X is a solution of the Sylvester equation with the scaling factor γ

$$S_{11}X - XS_{22} = \gamma S_{12} \tag{69}$$

and the equation (68) is a QR decomposition. Choice of the Sylvester equation with the scaling factor helps prevent possible calculation overflows.

When the Q matrix is obtained, it is applied to the submatrix S, and the entries of the $Q^T SQ$ that are supposed to become zero are checked. In case they are not small enough, it is assumed that the eigenvalues of the blocks are too close and that rotation will produce large errors. This means that the matrices of the Riccati equation are ill-conditioned.

If the Riccati equation is well-conditioned the Q matrix can be multiplied to the columns and rows of the matrix T that intersect with the two blocks intended for swapping. Matrix Q is also multiplied with the same columns of the orthogonal matrix U.

After the swapping is done, the 2x2 blocks might not be in the standard form anymore, so they need to be standardized. Standardization is done with the Givens rotations which are again applied to the columns of matrices T and U and rows of the matrix T intersecting the blocks that were swapped.

Sylvester equation is solved with the Gaussian Elimination with Complete Pivoting and scale factor is set accordingly if matrix entries with too small values are found.

7.3.2 Lyapunov equation solver

Lyapunov equation is just the special case of the Sylvester equation. Thus the solver of the Sylvester equation can be used. On the other hand, the specifics of the Lyapunov equation can be exploited for getting the solution with less computational effort. The implemented algorithm for solution of the Lyapunov equation uses one of these specialized algorithms, still it needs a regular Sylvester solver.

Consider the Lyapunov equation written as

$$\hat{A}^T \hat{X} + \hat{X}\hat{A} = \hat{C} \tag{70}$$

First step is to find a transformation of the matrix $\hat{A}$ to a quasi-triangular form. Quasi-triangular form allows the matrices to have subdiagonal elements with non-zero values. Real Schur form is quasi-triangular and can be provided with *schur* function of Meschach.

$$\hat{A} = UAU^T \tag{71}$$

U is the orthogonal matrix, and A is a real schur form of the matrix $\hat{A}$. The matrix U is then used to transform the matrix $\hat{C}$ too. That way we have following matrices.

$$\begin{aligned} A &= U^T \hat{A} U \\ C &= U^T \hat{C} U \end{aligned} \tag{72}$$

And the transformed Lyapunov equation is then

$$A^T X + XA = C \tag{73}$$

After the solution to the transformed equation is found, the final solution is obtained by reversing the transformation on matrix X with the matrix U, thus having

$$\hat{X} = UXU^T \tag{74}$$

To calculate the solution of the transformed Lyapunov equation algorithm based on the Bartels and Stewart is used. If the matrices A, C and X are partitioned like

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, C = \begin{bmatrix} C_{11} & C_{12} \\ C_{12}^T & C_{22} \end{bmatrix}, X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix}, \tag{75}$$

with A_{11}, C_{11} and X_{11} having dimensions of 1x1 or 2x2, depending on the dimension of the diagonal block, then the Lyapunov equation can be rewritten by three equations

$$A_{11}^T X_{11} + X_{11} A_{11} = C_{11} \tag{76}$$

$$A_{11}^T X_{12} + X_{12} A_{22} = C_{12} - X_{11} A_{12} \tag{77}$$

$$A_{22}^T X_{22} + X_{22} A_{22} = C_{22} - A_{12}^T X_{12} - X_{12} A_{12} \tag{78}$$

First equation (76) is of order 1 or 2, so the solution of X_{11} is trivial in the first case, and in second case it can be rewritten as a linear system of order 3.

$$\begin{bmatrix} a_{11} & a_{21} & 0 \\ a_{12} & a_{11} + a_{22} & a_{21} \\ 0 & a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{22} \end{bmatrix} = \begin{bmatrix} c_{11}/2 \\ c_{12} \\ c_{22}/2 \end{bmatrix} \tag{79}$$

This system is then solved using *QRfactor* and *QRsolve* Meschach functions.

Second equation (77) is of the form of the Sylvester equation. To solve it, a separately implemented solver for Sylvester equation is used. The solution produced is also copied to a diagonally symmetrical positions to fill in the complete solution of the Lyapunov equation.

Lastly, the third equation (78) is another Lyapunov equation, only with reduced dimensions of matrices. This new Lyapunov equation can be recursively solved with this same algorithm.

Sylvester equation solver Bartels and Stewart method is employed for solving of Sylvester equation. The matrices of the equation need to be transformed in the triangular form before applying the method. As for the Lyapunov equation, Schur factorization is also suitable. However the matrices passed to this algorithm are from the solver of the Lyapunov equation and they will already be in a quasi-triangular form. So this step can be skipped.

In the book it is suggested that the solution can be found by traversing matrices A and B of the equation through the diagonal blocks and forming reduced order Sylvester equations.

$$\begin{aligned} A_{kk}^T X_{kl} + X_{kl} B_{ll} &= C_{kl} - R_{kl} \\ R_{kl} &= \sum_{i=1}^{k-1} A_{ik}^T X_{il} + \sum_{j=1}^{l-1} X_{kj} B_{jl} \end{aligned} \tag{80}$$

where k and l denote the selection of submatrices by row and column indexes. Depending on the dimensions of diagonal blocks in A and B, they select the number of rows and columns respectively. For 1x1 block in matrix A, k will select one row or column, and for blocks of size 2x2, two rows or columns will be selected. Same applies for matrix B and l. Selected submatrices on the diagonal of A and B do not intersect.

These reduced Sylvester equations can be rewritten to the linear algebraic equations of the orders 1, 2 or 4. The linear systems can be written as

$$(I_{\hat{l}} \otimes A_{kk}^T + B_{ll} \otimes I_{\hat{k}})vec(X_{kl}) = vec(C_{kl} - R_{kl}) \tag{81}$$

here $\hat{l}$ and $\hat{k}$ are the numbers of the selection of rows or columns by l and k. Once again, the linear system of equation is solved with *QRfactor* and *QRsolve* Meschach functions.

Once all of the reduced Sylvester equations are solved, those solutions can be combined for a solution of the original Sylvester equation.

8 Testing and verification

Software developed for the onboard computer needs to be througly checked and verified before it is considered valid for the mission. This have been done in two ways. First, corecctness of the algorithms for solving of the AMRE were checked by comparing their outputs with the output of Matlab's AMRE solver. Second, verification have been done with the Hardware In the Loop testing method.

8.1 Riccati solvers numerical precision

In order to check the precision of the implemented algorithms for solving of AMRE, they were compared with the solver provided by the Matlab. The algorithms were compiled in to the Matlab's binary executables and they were called alongside the Matlab's solver in the simulations, providing them all with the same input matrices. Formula used for the measurement of the precision error is

$$e = \left\| \frac{X_1 - X_2}{\|X_1\|_F} \right\|, \ \|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \tag{82}$$

X_1 in the equation represents the solution of the AMRE from the Matlab's built-in function, while X_2 is a solution from one of the implemented algorithm combinations.

Three combinations of the algorithms were tested. First combination uses only Schur decomposition algorithm. Second combination is a Schur decomposition with refinement of the result by the Kleinman algorithm for every calculation of the control signal. And the third is the Kleinman algorithm with reuse of the result from previous calculation as the starting guess. Third combination also needs the Schur decomposition algorithm to produce the first guess, or to renew the guess whenever the Kleinman algorithm takes more than 5 iterations to converge. Test data was collected during the LQR and SDRE simulation tests in the Chapter 6. The control signal update frequency in those tests is 1 Hz, which amounts to 360000 test data samples. Table 2 contains the mean absolute errors, standard deviations and the percentage of the correct solutions during the test. Solutions are regarded as correct if the precision error is less than 0.1.

combination	mean error	standard deviation	correctness
1	2.1557	285.6	94.73%
2	4.0377	1081.3	95.76%
3	4.1014	647.8	81.95%

Table 2: AMRE solving precision

From the table it can be seen that the first combination have the smallest average error, and they vary significantly, but the least among the combinations. Its percentage of correct solutions is quite high. Second combination improves the number of correct solutions slightly over the first one, and the mean error and standard

deviation are increased. Standard deviation is especially high. Third combination have the greatest average errors. The number of correct values is the lowest. Generally the errors are quite small most of the time for all three combinations, but occasionally really big errors can happen. Second combination is selected for the use because of the highest percentage of correct solutions.

8.2 Hardware In the Loop testing

Developing of the control methods and algorithms in the Matlab is convenient as the environment is quite flexible and plenty of built-in functions are available. However, the satellite's onboard computer provides significantly different environment, and during the transfer process of the designed control algorithms from Matlab, it is easy to introduce errors. Also, the onboard software consists of several different modules, and all of them need to work properly at the same time. Thus, a mean to verify correct functionality is very valuable. Hardware In the Loop testing is a method which can provide the needed validation of the software and the hardware modules.

HIL testing is possible with the real-time target machine, which is to provide the interface between the tested hardware module (onboard computer) and the simulation environment. This way the onboard computer, with its software, can operate as during the mission, and through the real-time target machine emulation of the sensor readings and reactions to the actuators in the orbit environment is achieved. Figure 36 depicts the HIL testing setup.

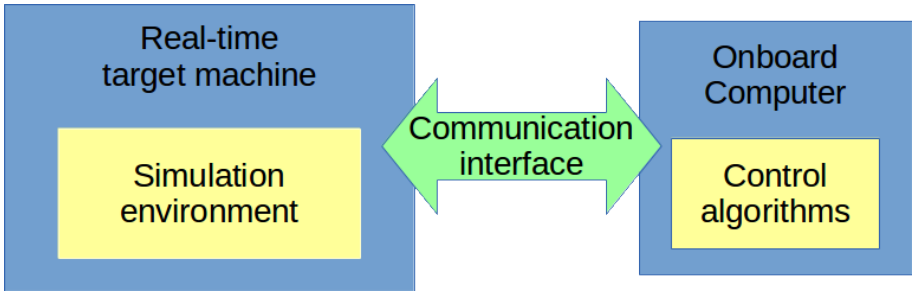

Figure 36: Hardware In the Loop setup

Real-time target machine used for the testing is produced by the Speedgoat GmbH. It contains the configurable FPGA I/O module, with which it can provide different hardware communication interfaces. Simulation environments supported on this machine can be generated with Simulink. This enables the use of the same simulation environment used for the computer simulations of the control methods for the HIL testing as well. This is important aspect, as the possibility to include new errors is avoided, and the whole process of setting up of the tests is simplified.

Communication interface used between the onboard computer and the real-time target machine is I^2C. This communication protocol is often used in the embedded systems, and is supported by great number of ICs, sensors and actuator driver boards.

8.2.1 AMRE algorithms timings

The control methods need to fulfill timing requirements to be effective. That is, the control signal is valid, not only if the solution from the algorithm is precise, but also if the solution is provided in specified time. Solving of the AMRE is considered as a computationally intensive problem, so there is a need to verify if the solution can be available in time.

The control signal update frequency was selected to be 1 Hz, so the solution of the AMRE should be provided at least once every second. Because the onboard software should have other tasks beside the ACS related tasks, the available time fraction to the control is set to the 40%. This means that the AMRE solvers need to provide the solution in less than 400 milliseconds.

Implemented AMRE solvers were run on the onboard computer, and the times needed to generate the solutions were recorded. Schur decomposition is a direct method and the time needed is somewhat constant over all of the solution calculations. On the other hand, Kleinman is iterative method and the time needed for finding of solution is nearly proportional to the number of iterations.

Average computation time for the Schur decomposition method is 16.93 milliseconds. And Kleinman's average iteration time is 4.27 millisecond. Average number of Kleinman iterations for second algorithm combination is 3.5, that is 14.95 milliseconds per solution. Finally, summing the two times gives approximately 32 millisecond for finding of the solution for the AMRE. This average needed computation time is more than 10 times less than the available time. Before concluding that the timing requirement is met, the worst case is also investigated. Largest recorded times for Schur decomposition and Kleinman algorithms are 33 and 55 milliseconds, respectively. Those times were not recorded for the same control signal calculation, though even if they were, calculation time budget would not have been exceeded. Thus, the developed AMRE solvers are able to provide the solutions within required time.

8.2.2 Overall functionality test

Software on the onboard computer consists of several tasks, every with its own specific job and dependence on other tasks. Tasks are mutually synchronised through the events and message queues, transmiting signals and exchanging data. Proper working of every individual task is important, but the complete system functionality depends also on the proper use of inter task communication. Complex multi-task structures can introduce possibilities for lock ups and problems that do not exist in a single task structures. This kind of problems can mostly be avoided by following certain software design guidelines. However, to be certain that there are no major problems in the system, throughout functionality testing is needed.

Another concern is the differences in the data provided to the onborad computer from the data provided to the control methods in the computer simulation, and also the difference of the control signals provided by the onboard computer in regard to the control signals from the computer simulation. In computer simulation, data is represented in 64-bits floating point format. However, data is converted to the

16-bit floating point format, for transfer through the I^2C communication interface. And the control signal is converted too from 64-bits to the 8-bits format. This loss of the precision is anyway expected, as the ADS have a limited precision on its own. The onboard computer also turns off the control signal for short time periods to read out the values from magnetometers. Though these changes are not significant, they can alter the satellite's behaviour during stabilization. This is another reason to subject the realized ACS to the HIL testing.

Such check was performed with the detumbling and LQR controllers, both at the orbit with altitude of 380 km. Satellite's attitude, angular velocity and magnetic dipole from the magnetorqueres were recorded. Figure 37 shows the angular velocities of the satellites body, with the gain equal to 15000. Comparing with the figure 20 there can be seen only small differences. This proves the validity of $\dot{B}$ implementation on the onboard computer and of the system's tasks.

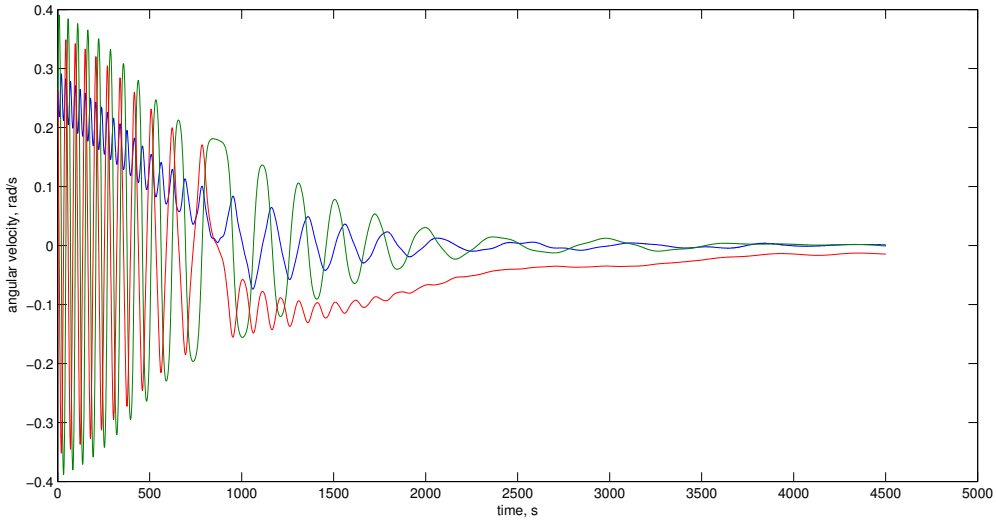

Figure 37: $\dot{B}$ HIL, angular velocity

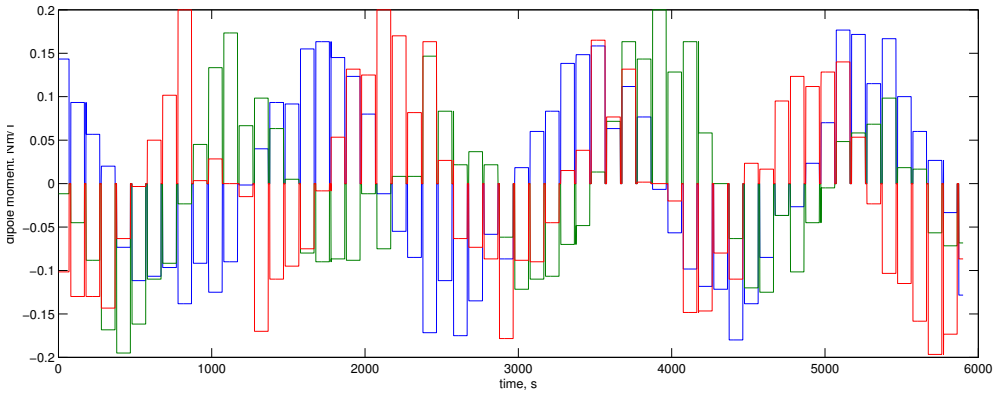

Figure 38: $\dot{B}$ HIL, control signal

On the figure 38 is shown how the control signal looks, with the off periods during the magnetometer sensors reading. On the figure 39 is the plot of the satellites attitude in unit quaternion during the LQR control. Successful control means that the AMRE solvers are providing correct solutions on the onboard computer. Finally, figure 40 contains the plot of the control signal generated during LQR test, where most of the time control is within the saturation level. Spikes on the plot are the occurances of the big errors in AMRE solutions.

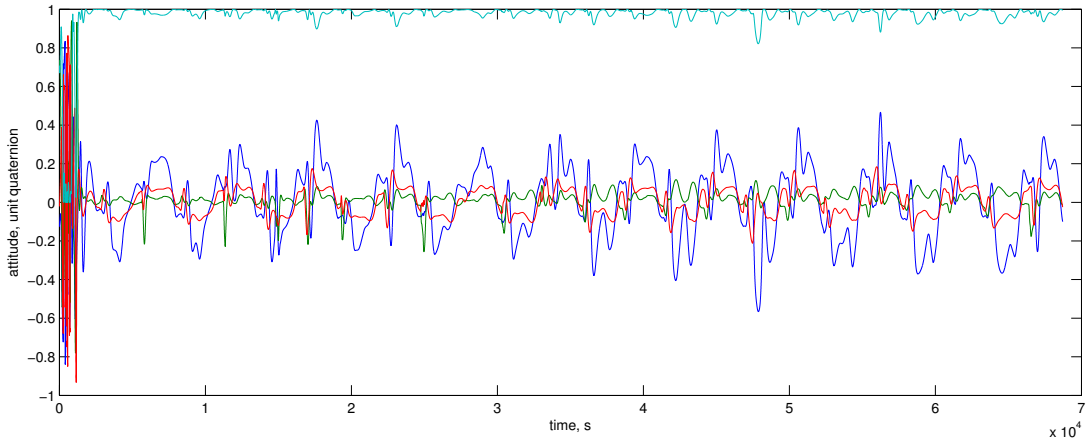

Figure 39: Attitude unit quaternion of LQR HIL test

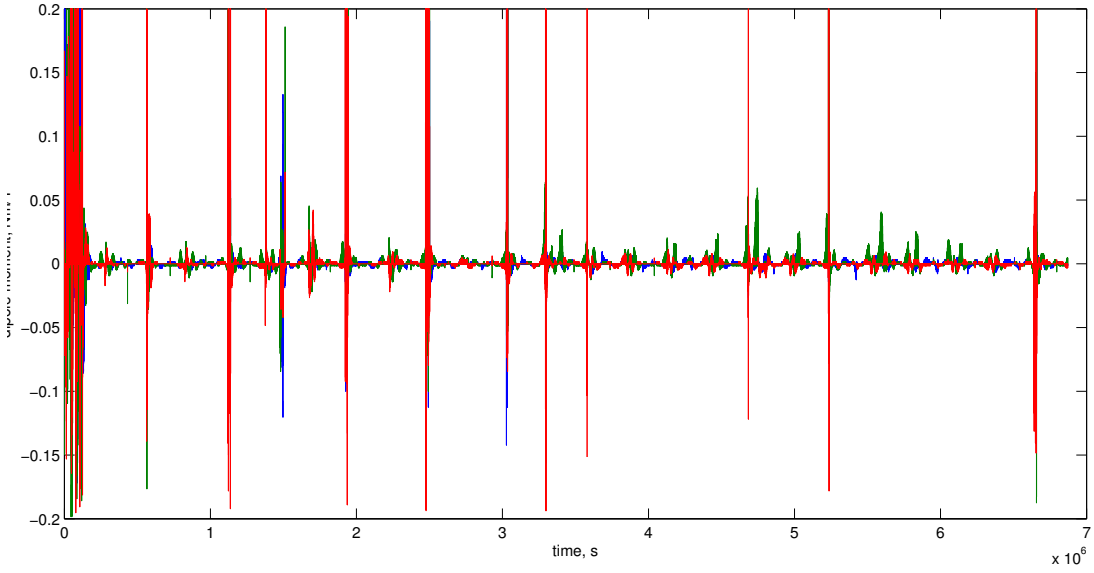

Figure 40: Dipole moment of LQR HIL test

9 Summary of findings

9.1 Discussion

Multiple control methods were investigated and tested both with computer simulations and Hardware In the Loop testing. First, the passive atmospheric drag torque stabilization was explored. Passive stabilization on its own is inadequate for stabilization in desired attitude, as the satellite's body design options are limited. However, it is possible to enhance to a lesser extent the tendency of the body toward the desired attitude. Still on the lower altitudes, the torque becomes really strong, which introduces strong oscillations. Also, as the modeling of the atmospheric drag forces is complex, and it is not easy to simulate them precisely, relaying on them is risky. Active method of control is required to attain stability.

Detumbling controller have satisfied the requirements needed for reducing of the initial angular velocity. Three gains were tried and all of them reach sufficiently low angular velocity in a short period of time. $\dot{B}$ proved to be a good choice for detumbling phase of satellite.

PD controller is the simplest approach examined for the nominal phase. However, it showed the poorest performance, with often angle errors greater than 15°. Stronger oscillations were detected in a lower altitude test. Reason for poor performance is that the controller is unaware of the environmental disturbances. Control signal produced by the controller must not be too strong, for if the control torque and the strong disturbances align in the same direction, the satellite will start tumbling. Thus the control signal have to be low, though in this situation the disturbance torques can dominate and destabilize the satellite's attitude.

LQR controller showed the best stabilization ability among the tried controllers, though it still can not fully satisfy the mission requirement. The satellite, under LQR control, can attain low angle errors while there is at least some control over all of its body axes. Larger angle errors happen when y or z axes completely loose the controllability due to the orientation of the magnetic field vector. At those moments, disturbance torques can directly and with full strength affect the angle error. Also it is noticed that the LQR is less effective while the attitude is far from the desired state, making the LQR insufficient, as the period until the satellite reach the vicinity of the goal state might be long.

SDRE controller uses the nonlinear model, which is an improvement over LQR when the state is far from the goal state. SDRE is able to quickly reach the region near the desired attitude. However, SDRE showed somewhat worse performance for closely tracking of refernce point. Problem with controllability of the LQR also appears for the SDRE as well.

Software realization for the onboard computer performed very well with the HIL testing, within the performance results of the computer simulations. Timing requirements are met, and precision of the solutions is mostly adequate.

All the tests were done with ideal values of the system state. This is not realistic and with the proper ADS subsystem integrated together with the ACS better analysis can be performed. Slightly less precise stability is expected as the effect of

the imprecisions introduced by the ADS in the provided system state values.

9.2 Further work

Presented control methods were able to reach certain level of attitude stability, though they do not completely satisfy the defined requirements. Some ideas are presented as possibility to improve the stability performance.

The biggest problem for the stability are high disturbance torques during the loss of controllability over one of the axes. Addition of the momentum wheel to the satellite, aligning its axis of rotation with the satellite body's y axis, will add rigidity to the x and z axes. This would make those axes less susceptible to the disturbances. However there is a possibility that this will create additional burden to the y axis, creating greater angle errors around it. Still, the overall performance might improve, increasing the percentage of time spent with angle error less than 15°. Additionally, momentum wheel could lessen the effect of the noises on the sensors and system state values.

Even greater improvement might be achieved if the momentum wheel could also act as a reaction wheel when the controllability of the magnetorquers is lost over the y axis. Unfortunately, this would greatly complicate the overall system and might need significant control adjustments.

For the future work it is unavoidable to integrate the ACS and ADS together in the same system. Complete and thorough testing will be required to verify the system after the integration.

9.3 Conclusion

Attitude control system with passive atmospheric drag torque stabilization and active magnetorquer control was analyzed with different control methods. Implemented control methods are $\dot{B}$, PD, LQR and SDRE controllers. Computer simulations showed that the LQR performed the best, though achieved stability does not satisfy the satellite mission requirements completely. Some suggestions to improve the stability performance were given.

Software for the onboard computer was developed. Tasks of the system were defined and algorithms for control methods were implemented. The whole system was tested with the Hardware In the Loop method. The software proved to be correct and valid, satisfying all functionality, timing and precision needs.

References

Meschach libray web site, june 2014. URL http://homepage.math.uiowa.edu/~dstewart/meschach/.

M. Abdelrahman, I. Chang, and S.-Y. Park. Magnetic torque attitude control of a satellite using the state-dependent riccati equation technique. *International Journal of Non-Linear Mechanics*, 46(5):758–771, 2011. cited By (since 1996)7.

Zhaojun Bai and James W. Demmel. On swapping diagonal blocks in real schur form. *Linear Algebra and its Applications*, 186(0):75 – 95, 1993. ISSN 0024-3795. doi: http://dx.doi.org/10.1016/0024-3795(93)90286-W. URL http://www.sciencedirect.com/science/article/pii/002437959390286W.

S.P. Bhat and D.S. Bernstein. A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon. *Systems and Control Letters*, 39(1):63–70, 2000. cited By (since 1996)176.

Angelika Bunse-gerstner. Computational solution of the algebraic riccati equation.

C. Böhm, M. Lazar, and F. Allgöwer. Stability of periodically time-varying systems: Periodic lyapunov functions. *Automatica*, 48(10):2663–2669, 2012. cited By (since 1996)3.

CC Finlay, S Maus, CD Beggan, TN Bondar, A Chambodut, TA Chernova, A Chulliat, VP Golovkov, B Hamilton, M Hamoudi, et al. International geomagnetic reference field: the eleventh generation. *Geophysical Journal International*, 183 (3):1216–1230, 2010.

Peter Fortescue, Graham Swinerd, and John Stark. *Spacecraft systems engineering*. John Wiley & Sons, 2011.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

Felix R Hoots, Ronald L Roehrich, and TS Kelso. Spacetrack report no. 3. 1980.

Kasper Fuglsang Jensen and Kasper Vinther. Attitude determination and control system for aausat3. Master's thesis, Aalborg University, Denmark, 2010.

Sergey Katsev. Streamlining of the state-dependent riccati equation controller algorithm for an embedded implementation. 2006.

David Kleinman. On an iterative technique for riccati equation computations. *Automatic Control, IEEE Transactions on*, 13(1):114–115, 1968.

Alan J. Laub. Schur method for solving algebraic riccati equations. *IEEE Transactions on Automatic Control*, AC-24(6):913–921, 1979. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0018681625&partnerID=40&md5=8da62d68845da1409b3f38d76c491ae9. cited By (since 1996)280.

C.-L. Lin and C.-L. Chen. Realisation of a riccati equation-based controller using gradient-type neural networks. *Control Engineering Practice*, 9(3): 329–341, 2001. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0035282132&partnerID=40&md5=946d0e1c97913c16434fe4b7bebed3bf. cited By (since 1996)3.

M. Lovera, E. De Marchi, and S. Bittanti. Periodic attitude control techniques for small satellites with magnetic actuators. *IEEE Transactions on Control Systems Technology*, 10(1):90–95, 2002. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0036205172&partnerID=40&md5=87dbeb3375b5410acc27eaf77e0d9efb. cited By (since 1996)72.

C.G. Mayhew, R.G. Sanfelice, and A.R. Teel. On quaternion-based attitude control and the unwinding phenomenon. pages 299–304, 2011. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-80053149202&partnerID=40&md5=6df1b968b0a7496193d5e7a11982f41a. cited By (since 1996)9.

P.K. Menon, T. Lam, L.S. Crawford, and V.H.L. Cheng. Real-time computational methods for sdre nonlinear control of missiles. volume 1, pages 232–237, 2002. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0036060790&partnerID=40&md5=f22712d8216e0c09e0862abb1920cfb1. cited By (since 1996)22.

Riki Munakata et al. Cubesat design specification rev. 12. *The CubeSat Program, California Polytechnic State University*, 1, 2009.

B.M. Rao, M.K. Teja, and N. Nitin. Comparison of process scheduling methodologies for embedded systems. pages 387–391, 2009. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77949620468&partnerID=40&md5=4098217f404de6fea8c06ec855ef5202. cited By (since 1996)0.

Real Time Engineers Ltd. Freertos web site, june 2014. URL http://www.freertos.org.

VA Sarychev, SA Mirer, AA Degtyarev, and EK Duarte. Investigation of equilibria of a satellite subjected to gravitational and aerodynamic torques. *Celestial Mechanics and Dynamical Astronomy*, 97(4):267–287, 2007.

M.J. Sidi. *Spacecraft Dynamics and Control: A Practical Engineering Approach*. Cambridge Aerospace Series. Cambridge University Press, 1997. ISBN 9780521787802.

E. Silani and M. Lovera. Magnetic spacecraft attitude control: A survey and some new results. *Control Engineering Practice*, 13(3):357–371, 2005. cited By (since 1996)96.

Vasile Sima. *Algorithms for linear-quadratic optimization*, volume 200. CRC Press, 1996.

Fiona Singarayar, R. Reinhard, C. Asma, J. Thoemel, T. Scholz, C. Bernal, W. Weggelaar, G. Shirville, D. Kataria, and M. Richard. QB50 System Requirements, issue 5. Technical report, QB50, FP7 project, October 2013.

J.J.E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall International Editions. Prentice-Hall, Incorporated, Englewood Cliffs, N.J., 1991. ISBN 9780130400499. URL http://books.google.fi/books?id=HddxQgAACAAJ.

Jason L Speyer and David H Jacobson. *Primer on optimal control theory*, volume 20. SIAM, 2010.

Zdenko Tudor. Design and implementation of attitude control for 3-axes magnetic coil stabilization of a spacecraft. Master's thesis, Norwegian University of Science and Technology, Norway, 2011.

John Ting-Yung Wen and Kenneth Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic Control*, 36(10):1148–1162, 1991. cited By (since 1996)420.

J.R. Wertz. *Spacecraft Attitude Determination and Control*. Astrophysics and Space Science Library : a series of books on the recent developments of space science and of general geophysics and astrophysics. Reidel, 1978. ISBN 9789027709592.

Rafael Wisniewski. *Satellite Attitude Control Using Only Electromagnetic Actuation*. PhD thesis, Aalborg University, Denmark, 1996.

T. Yucelen, A.S. Sadahalli, and F. Pourboghrat. Online solution of state dependent riccati equation for nonlinear system stabilization. pages 6336–6341, 2010. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-77957818551&partnerID=40&md5=456e3788b42a78b25cb86d0e91477f25. cited By (since 1996)0.

A Mathematical operations

Unit quaternion inverse - $\mathbf{q}^{-1}$

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}, \mathbf{q}^{-1} = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix}$$

Quaternion multiplication - $\odot$

$$\mathbf{q}_a = \begin{bmatrix} q_{a1} \\ q_{a2} \\ q_{a3} \\ q_{a4} \end{bmatrix}, \mathbf{q}_b = \begin{bmatrix} q_{b1} \\ q_{b2} \\ q_{b3} \\ q_{b4} \end{bmatrix}$$

$$\mathbf{q}_a \odot \mathbf{q}_b = \begin{bmatrix} q_{b1}q_{a1} - q_{b2}q_{a2} - q_{b3}q_{a3} - q_{b4}q_{a4} \\ q_{b1}q_{a2} + q_{b2}q_{a1} - q_{b3}q_{a4} + q_{b4}q_{a3} \\ q_{b1}q_{a3} + q_{b2}q_{a4} + q_{b3}q_{a1} - q_{b4}q_{a2} \\ q_{b1}q_{a4} - q_{b2}q_{a3} + q_{b3}q_{a2} + q_{b4}q_{a1} \end{bmatrix}$$

Euclid norm - $\|\mathbf{r}\|$

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \|\mathbf{r}\| = \sqrt{r_1^2 + r_2^2 + r_3^2}$$

Vector operator - $vec(M)$

$$M = \begin{bmatrix} M_1 & M_2 & \dots & M_n \end{bmatrix}, M \in \Re^{m \times n}$$

$$vec(M) = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix}, vec(M) \in \Re^{m \cdot n}$$

Kronecker product - $\otimes$

$$A \in \Re^{m \times n}, B \in \Re^{p \times q}$$

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

Vector cross product - $\times$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

Skew symmetric matrix - $skew(\mathbf{r})$

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

$$skew(\mathbf{r}) = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}$$

Cross-product/skew-matrix equivalence

$$\mathbf{a} \times \mathbf{b} \equiv skew(\mathbf{a}) \cdot \mathbf{b}$$